

# Texture synthesis and image analogies



15-463, 15-663, 15-862  
Computational Photography  
Fall 2017, Lecture 9

# Course announcements

- Please take Doodle for second make-up lecture, link on Piazza.
- Homework 3 is out.
  - Due October 12th.
  - Shorter, but longer bonus component.

# Overview of today's lecture

- Reminder: non-local means.
- Texture synthesis.
- Texture by non-parametric sampling.
- Image quilting.
- Inpainting.
- Texture transfer.
- Image analogies.
- Deep learning teaser.

# Slide credits

Most of these slides were adapted from:

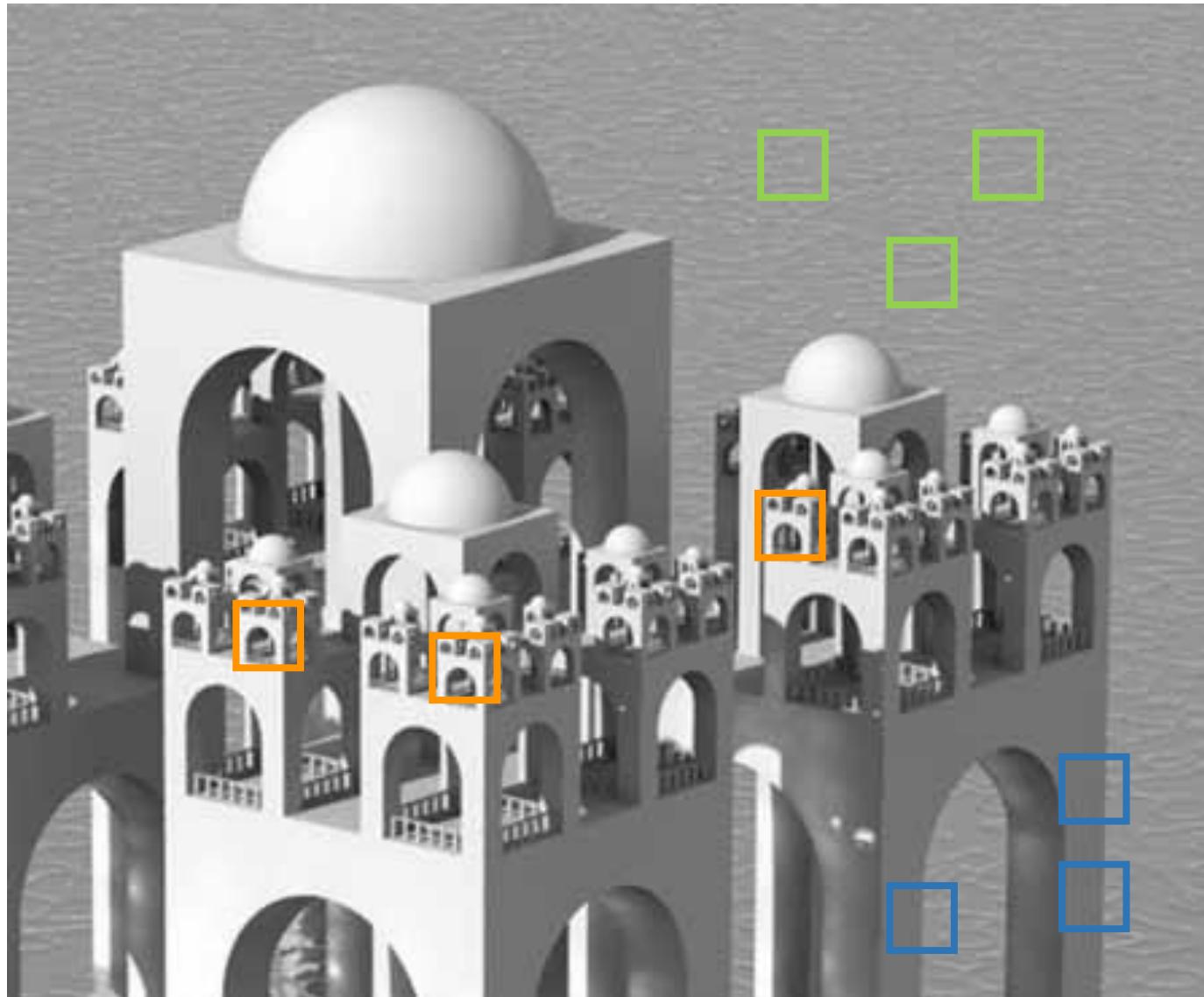
- Kris Kitani (15-463, Fall 2016).

Some slides were inspired or taken from:

- Fredo Durand (MIT).
- James Hays (Georgia Tech).

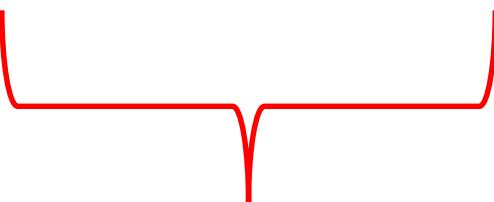
Reminder: non-local means

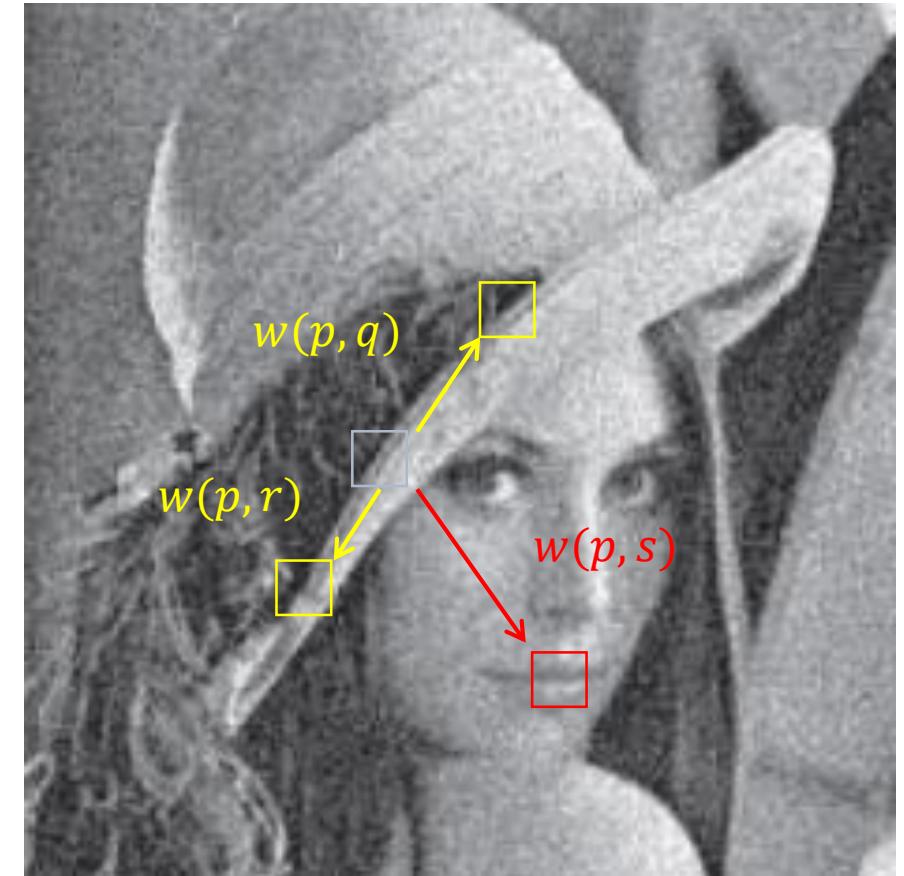
# Redundancy in natural images



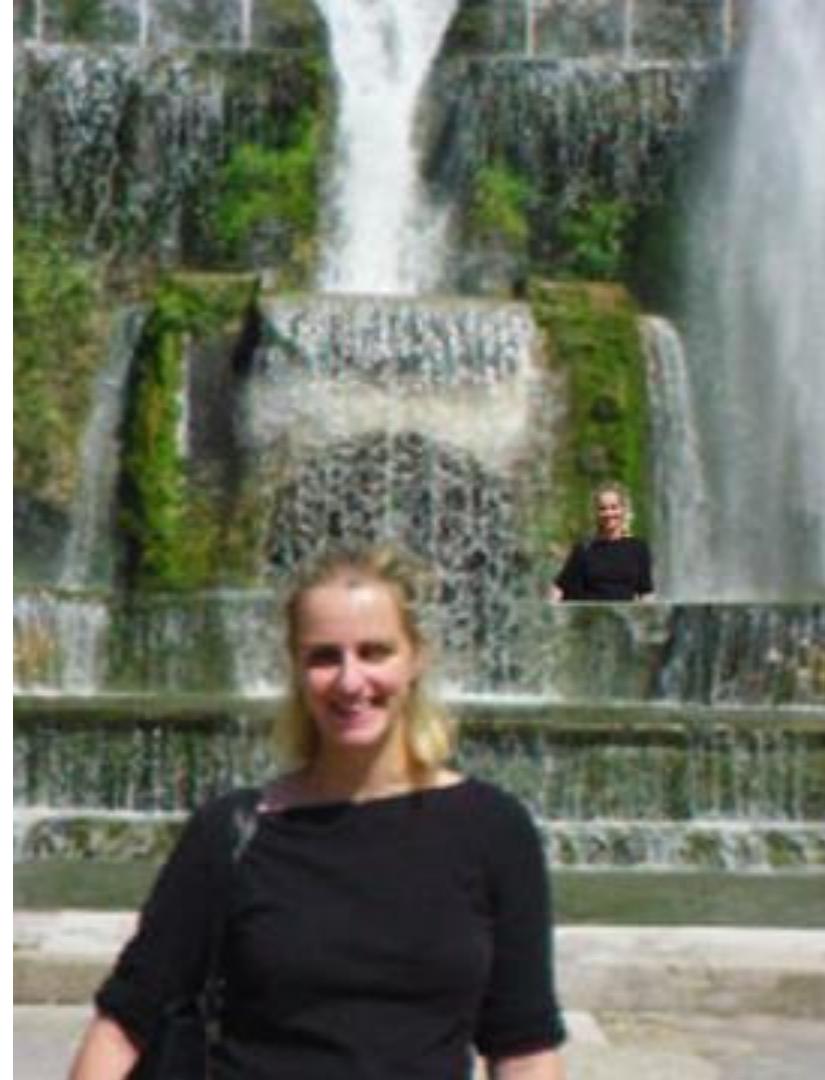
# Non-local means

No need to stop at neighborhood. Instead search *everywhere* in the image.

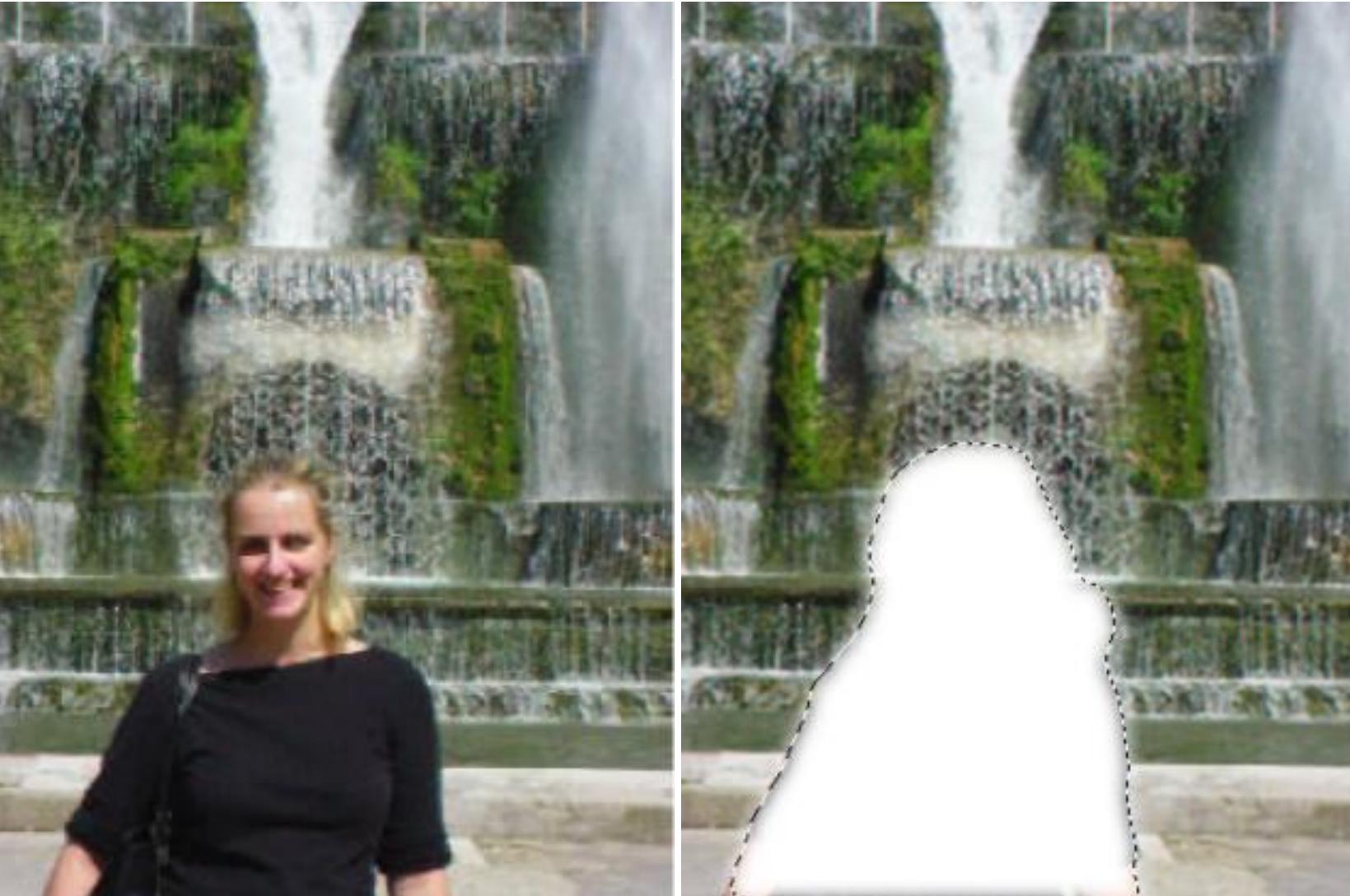
$$\hat{x}(i) = \frac{1}{C_i} \sum_j y(j) e^{-\frac{SSD(y(N_i)-y(N_j))}{2\sigma^2}}$$




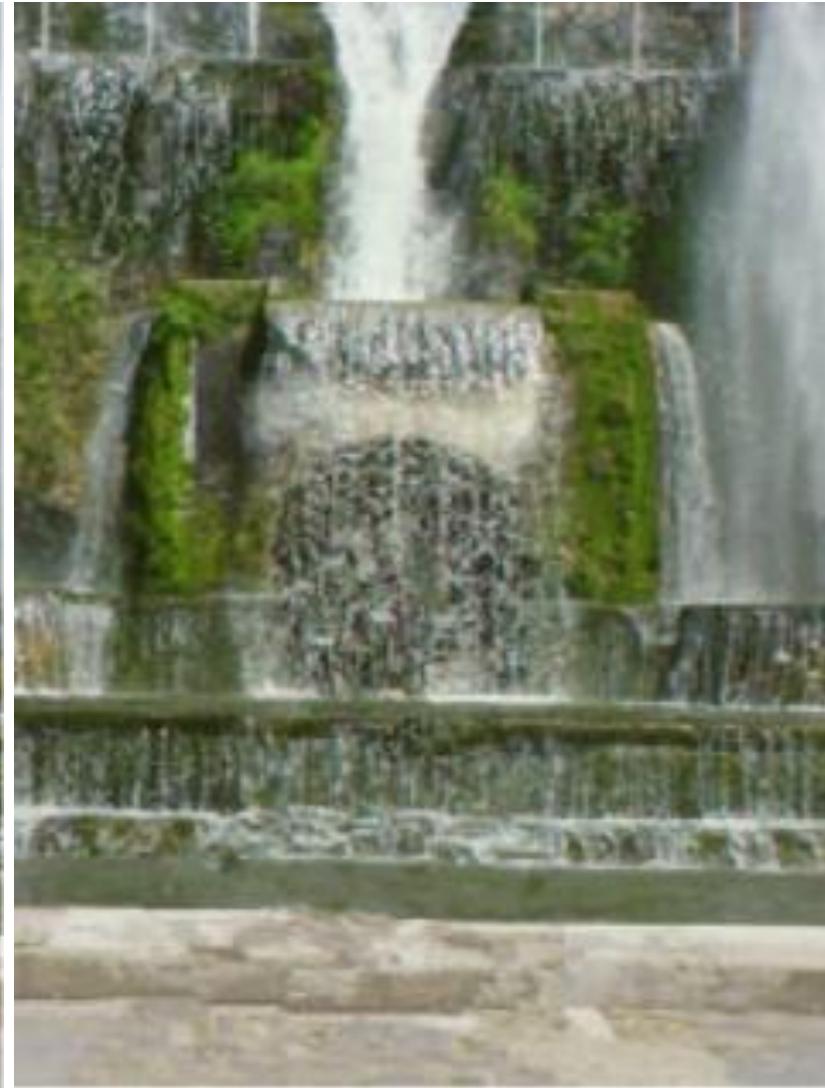
# Last couple of classes: adding things to the image



# This class: removing things from the image



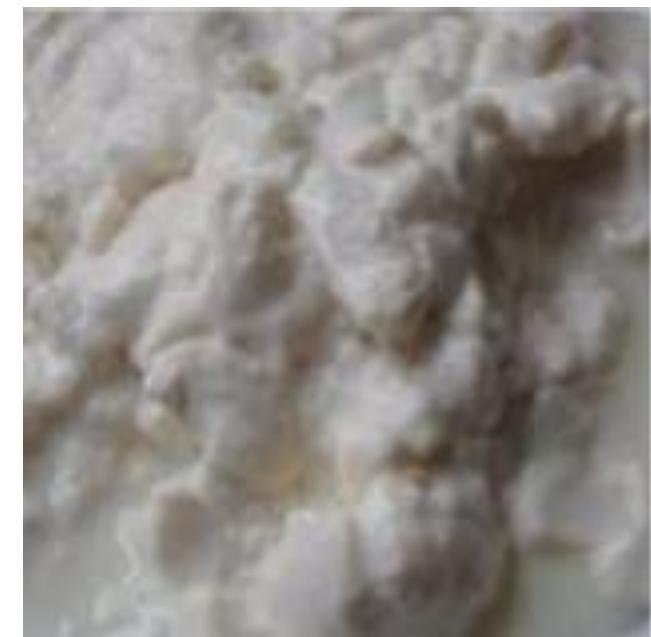
# This class: removing things from the image



# Texture synthesis

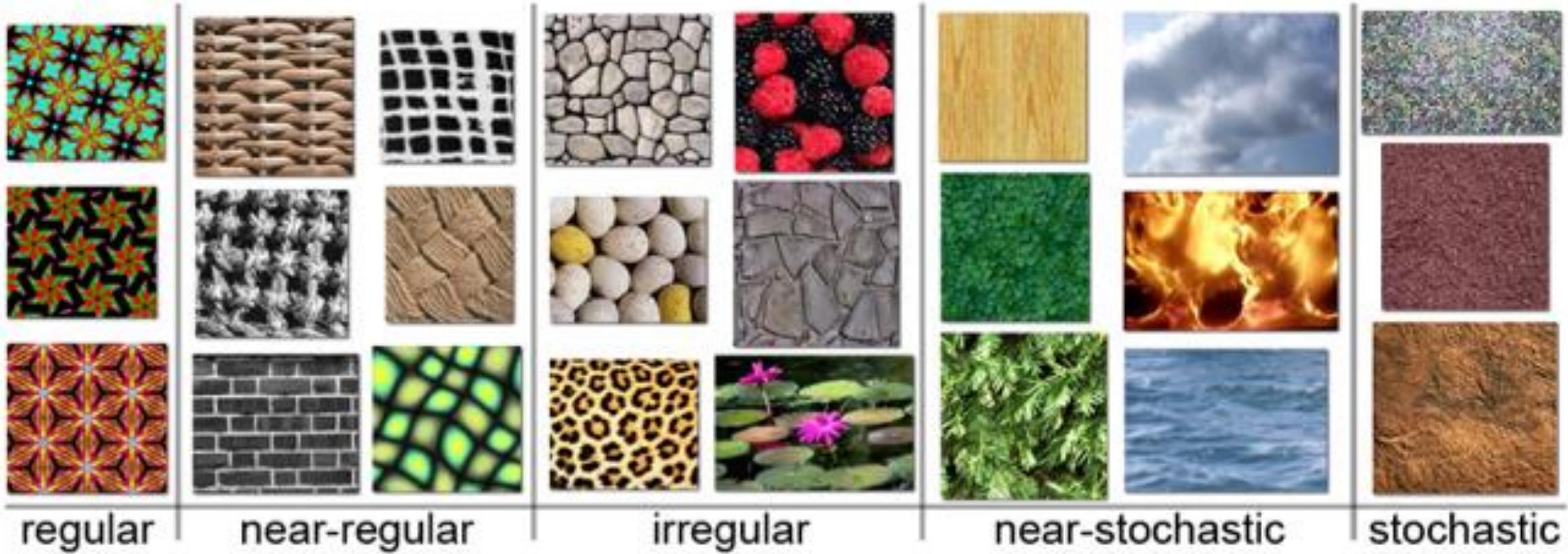
# Texture

- Depicts spatially repeating patterns
- Appears naturally and frequently



# Texture

- Large variety of textures



# Texture synthesis

Goal: create new samples of a given texture.

Applications:

- hole filling
- virtual environments
- view expansion
- texturing surfaces
- ....



# How would you do texture synthesis for this sample?

Input



# How would you do texture synthesis for this sample?

Input



tiling



random

# Approach 1: probabilistic modeling

Basic idea:

- Compute statistics of input texture (e.g., histogram of edge filter responses).
- Generate a new texture that keeps these same statistics.



Heeger and Bergen, "Pyramid-based texture analysis/synthesis," SIGGRAPH 1995  
Simoncelli and Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," ICIP 1998

# Approach 1: probabilistic modeling

Probability distributions are hard to model well.

input



output

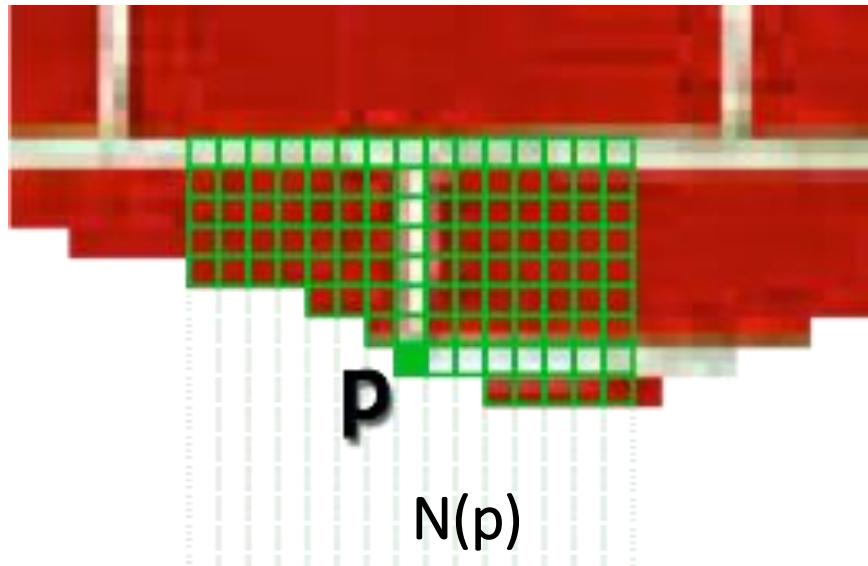


Any other ideas?

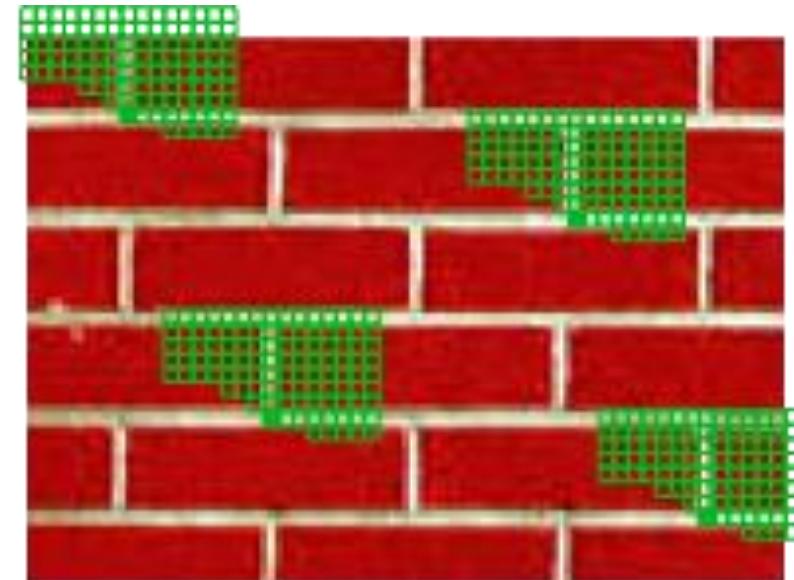
Texture by non-parametric sampling

# Approach 2: sample from the image

Run template matching, get  $N$  best matches, and sample one at random.



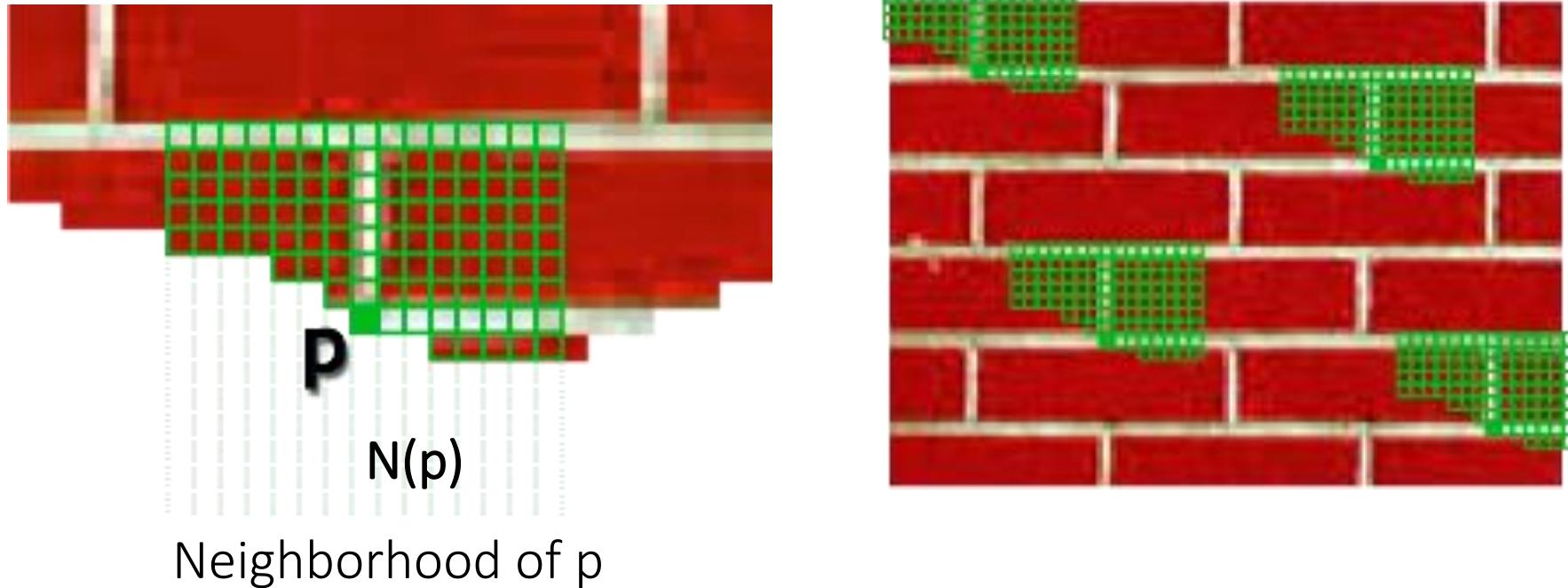
Neighborhood of  $p$



What are sampling from?

# Approach 2: sample from the image

Run template matching, get N best matches, and sample one at random.

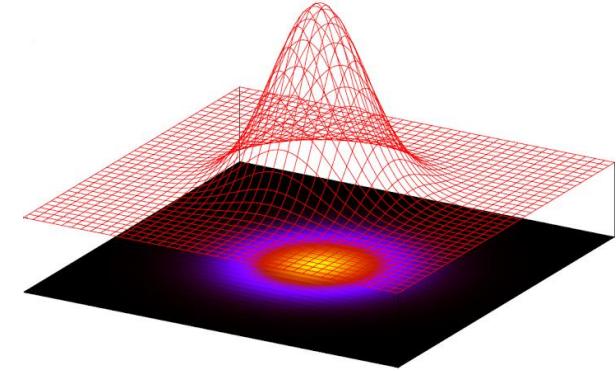


- Similar nearby images define a non-parametric PDF  $P(p|N(p))$
- By selecting a random sample, we are sampling from this PDF

# Implementation details

How do you define patch similarity?

# Implementation details

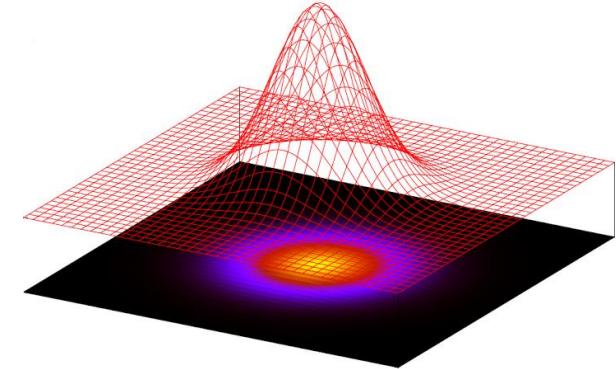


How do you define patch similarity?

- Gaussian-weighted SSD (emphasis on nearby pixels).

In what order should you synthesize?

# Implementation details



How do you define patch similarity?

- Gaussian-weighted SSD (emphasis on nearby pixels).

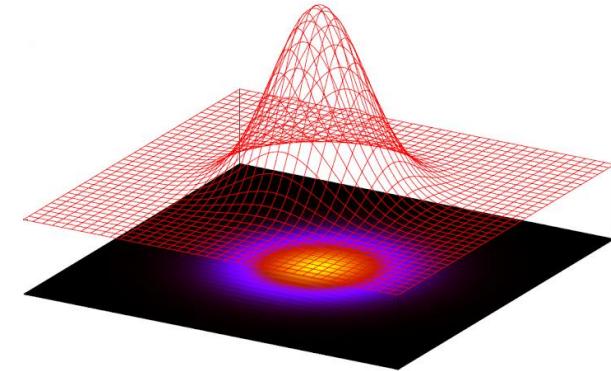


In what order should you synthesize?

- Onion-peel ordering – pixels with most neighbors are synthesized first.

How do you synthesize from scratch?

# Implementation details



How do you define patch similarity?

- Gaussian-weighted SSD (emphasis on nearby pixels).



In what order should you synthesize?

- Onion-peel ordering – pixels with most neighbors are synthesized first.



How do you synthesize from scratch?

- Pick a small patch at random from source.

# Ideas from information theory

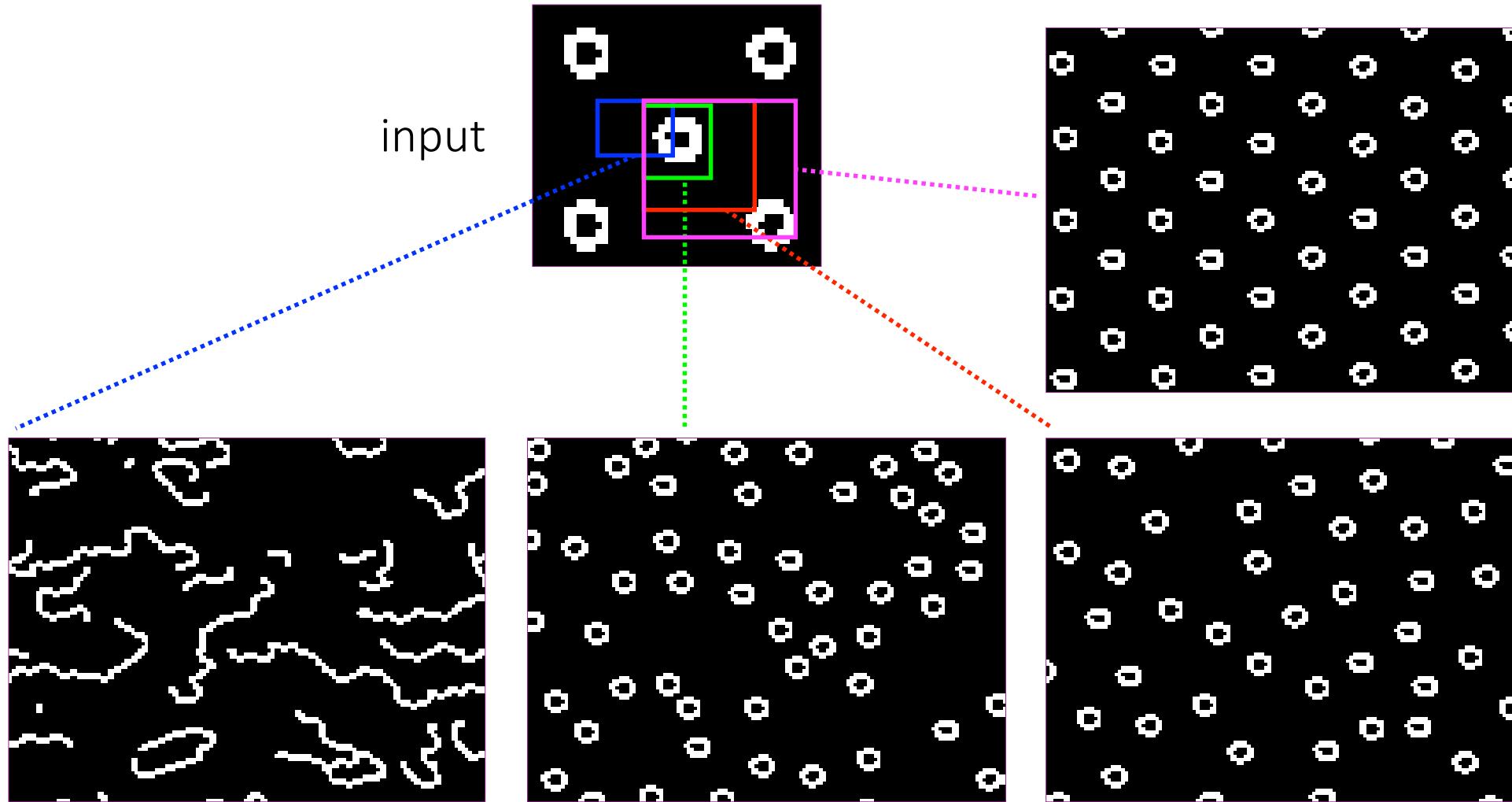
- Generate English-sounding sentences by modeling the probability of each word given the previous words (n-grams)
- Large “n” will give more structured sentences



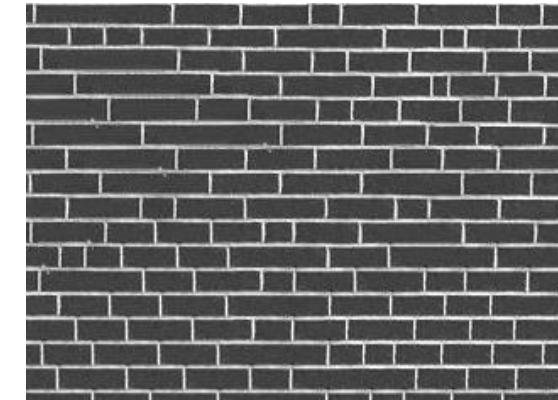
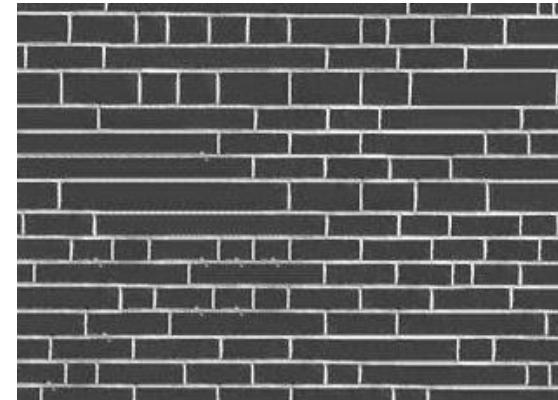
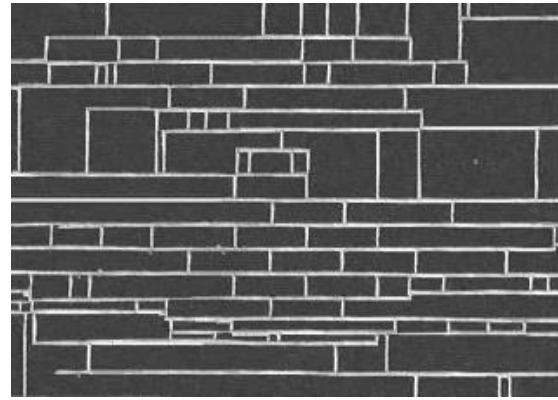
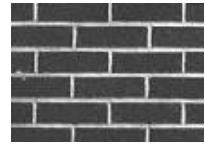
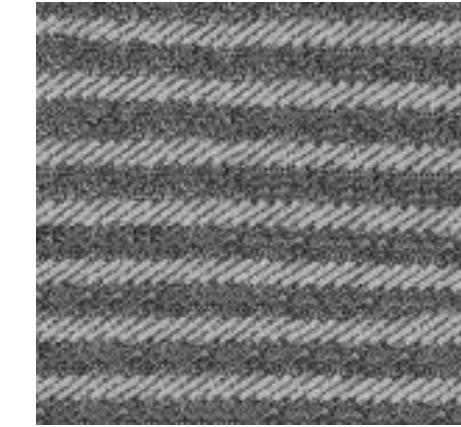
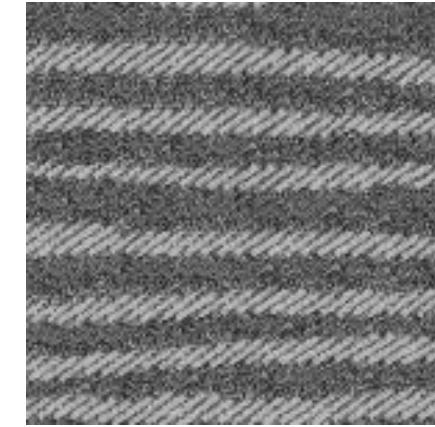
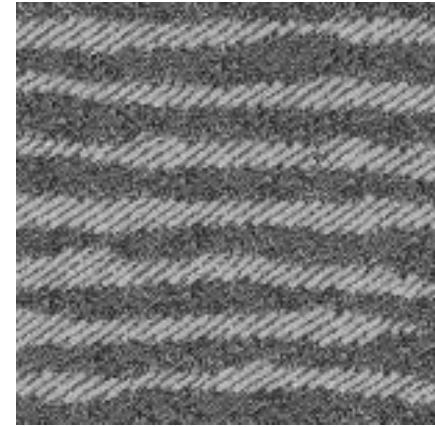
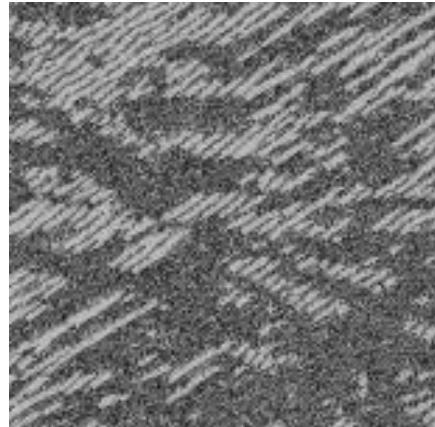
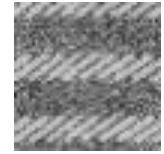
Claude Elwood Shannon  
(1916–2001)

*“I spent an interesting evening recently with a grain of salt.”*

# Size of neighborhood window matters a lot



# Size of neighborhood window matters a lot



patch size

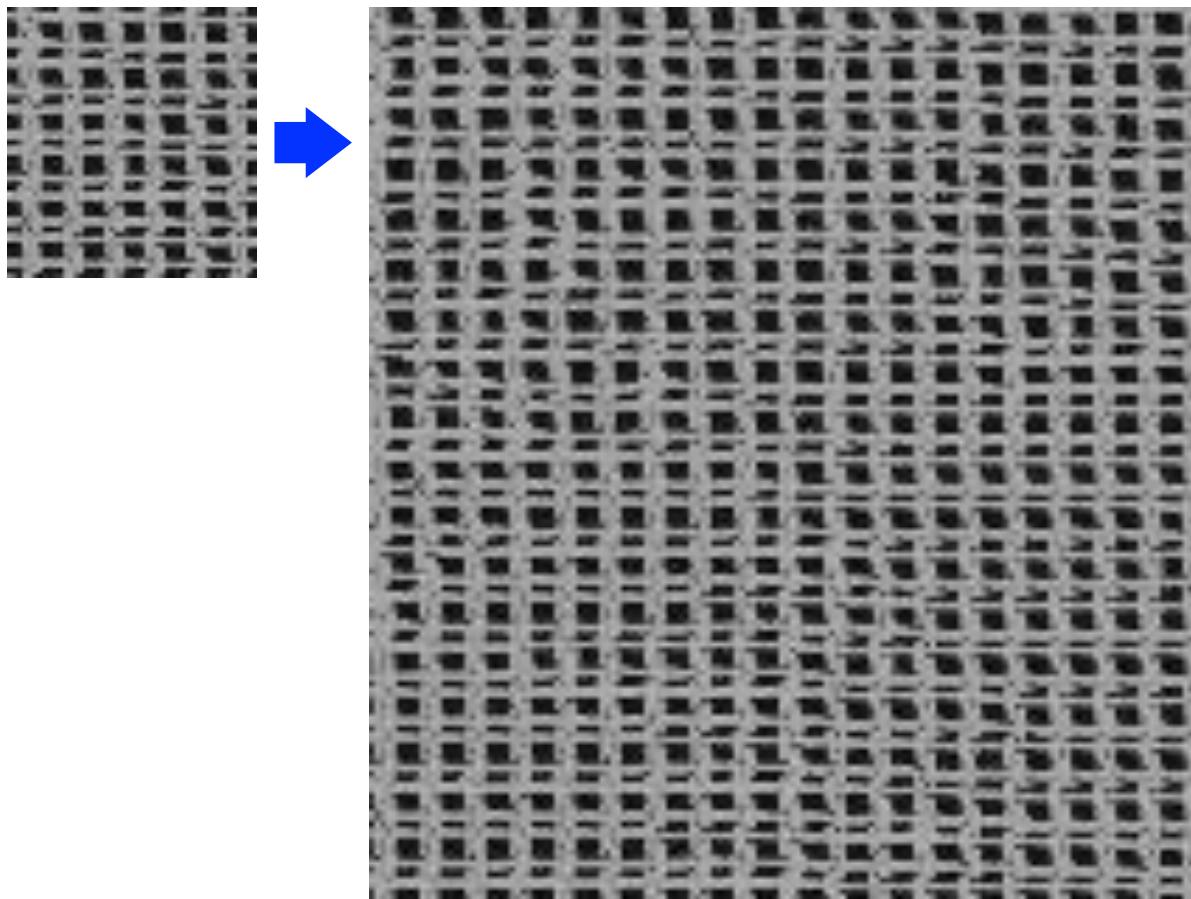
# Texture synthesis algorithm

While image not filled

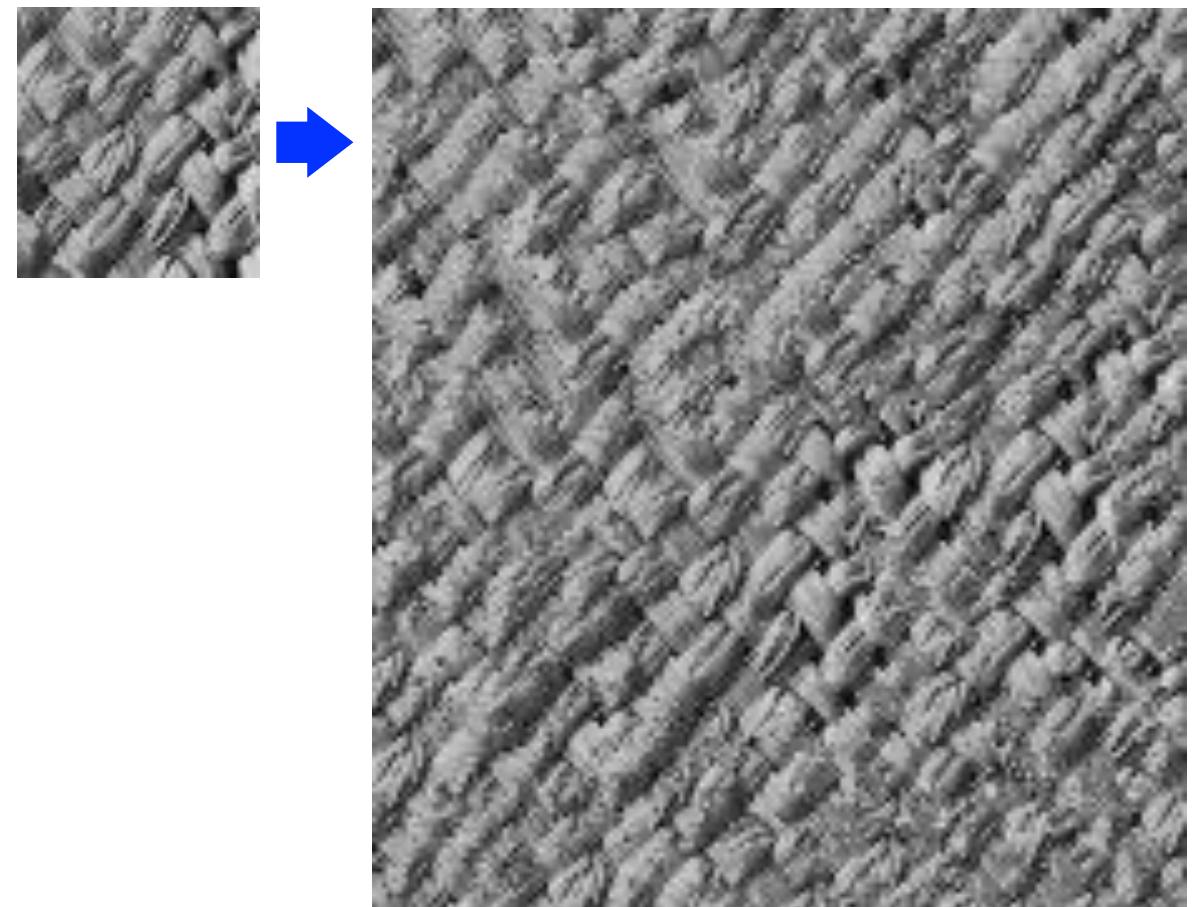
1. Get unfilled pixels with filled neighbors
2. Sort by number of filled neighbor
3. For each pixel
  - a) Get top N matches of visible neighbor  
(Patch Distance: Gaussian-weighted SSD)
  - b) Randomly select one of the matches
  - c) Copy pixel value

# Examples

French canvas

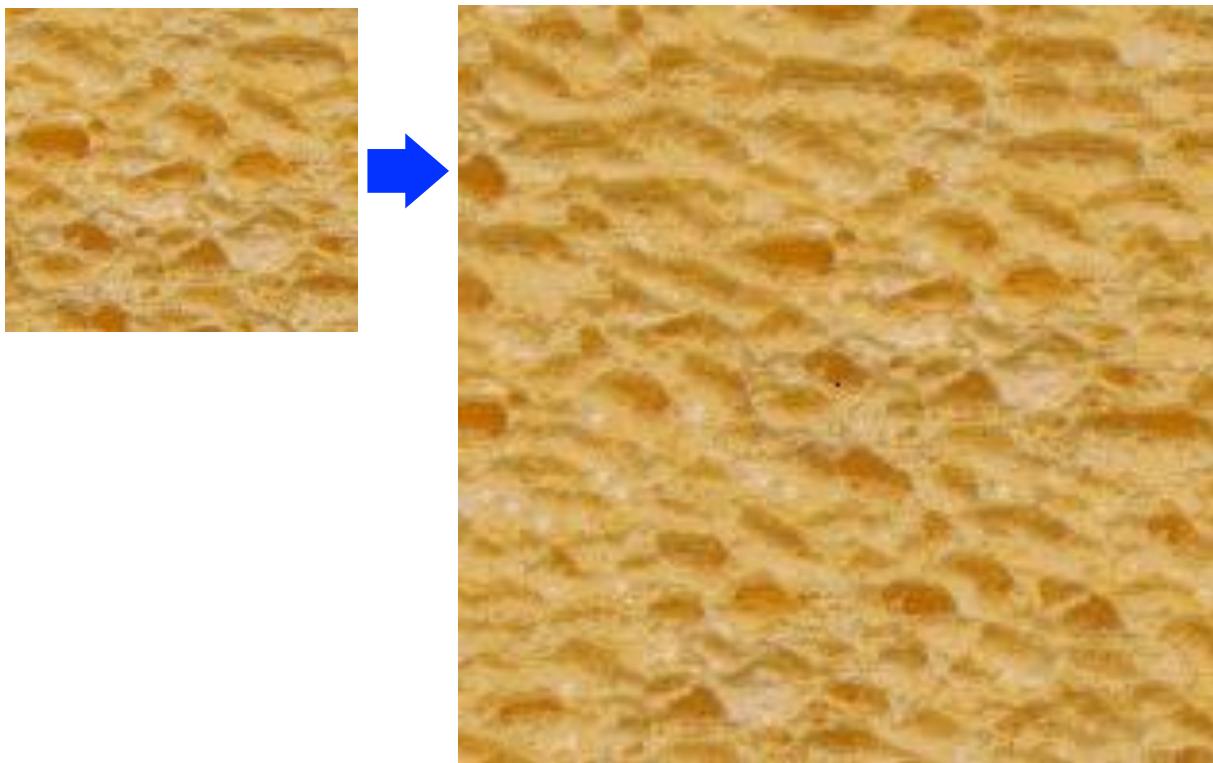


rafia weave

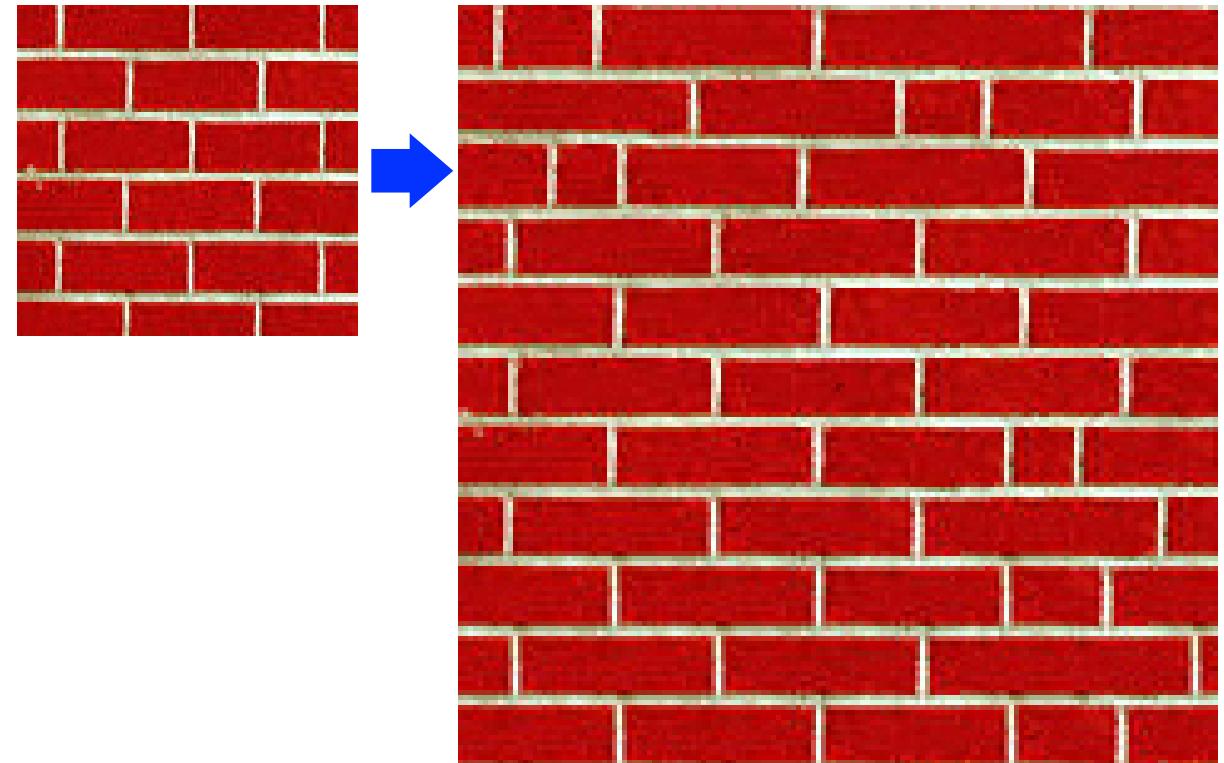


# Examples

white bread



brick wall



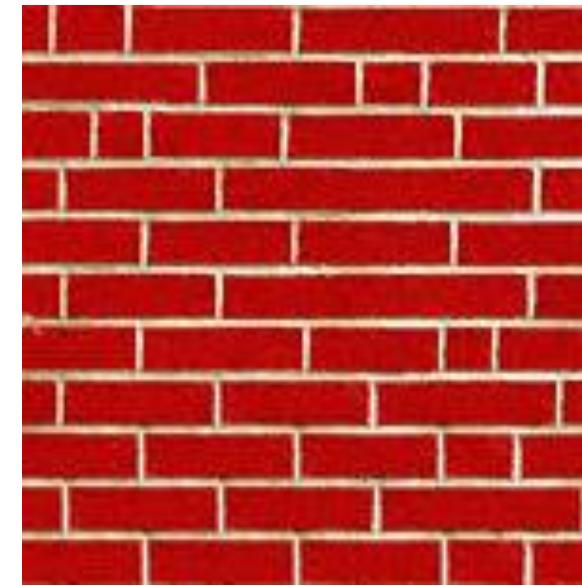
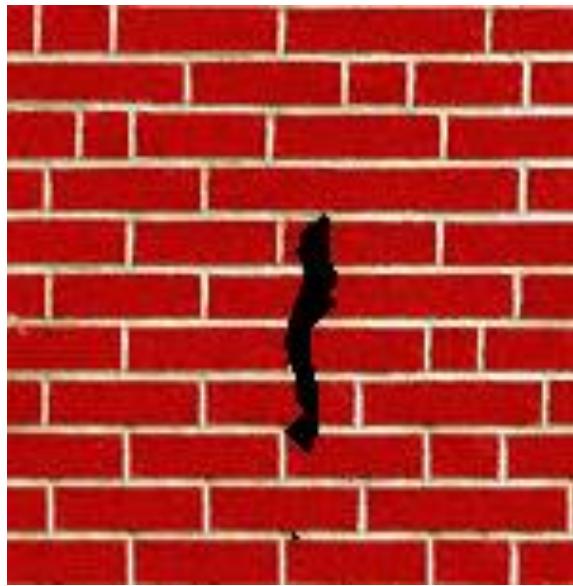
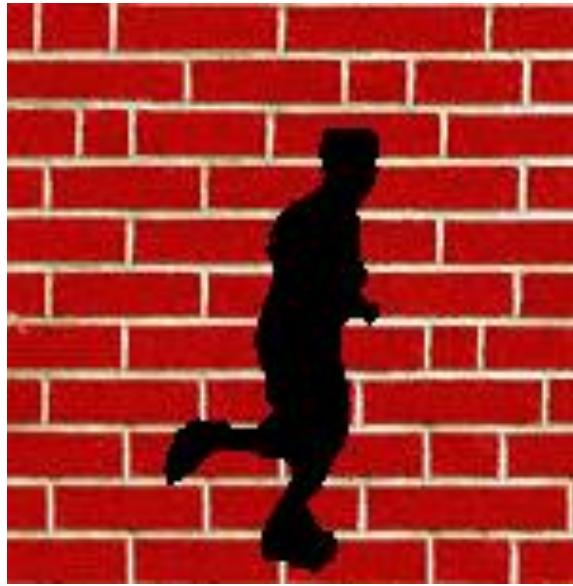
# Homage to Shannon

uring in the unsensational  
r Dick Gephardt was fai-  
rful riff on the looming :  
nly asked, "What's your  
tions?" A heartfelt sigh  
story about the emergen-  
es against Clinton. "Boy-  
g people about continuin-  
ardt began, patiently obs-  
, that the legal system k-  
e with this latest tanger

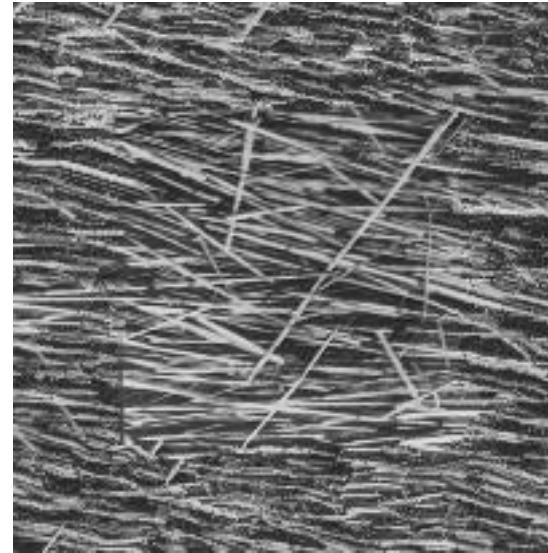
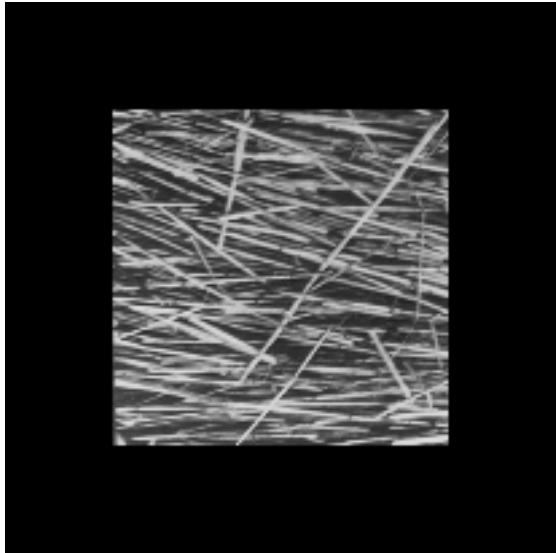


i thaim, them . "Wmephartfe lartifelintomimen-  
iel ck Clirtioout omaim thartfelins. f out s anetc  
the ry onst wartfe lck Gephtoomimeationl sigak  
Chiooufit Clinut Cll riff on, hat's yordn, panut tly:  
ons yoontonsteht wasked, paim i sahe loo riff on  
nskoneploourtfeas leil A nst Clit, "Wleontongal s  
k Cirtioouirtfepe óng pmé abegal fartfenstemem  
tiensteneltorydt telemephnišverdt was agemer  
ff ons artientont Cling peme asurtfe atih, "Boui s  
hal s fartfelt sig pedrildt ske abounutie aboutoo  
itfaone newwas yous aboronthardt thatins fain, ped,  
ains, them, pabout wasy arfiut cōuity d, l n A h  
ole emthringbooneme agas fa bontinsyst Clinut  
ory about continst Clipeopinst Cloke agatiff out C  
stome zinemen tly ardt beorabou n, thenly as t C  
cons faimeme Diontont wat coutlyohgans as fan  
ien, phrtfaul, "Wbaut cout congagal cōmininga:  
mifmst Cliuy abon al coountha.emungait tf oun  
Whe looorystan loontieph, intly on, theoplegatick t  
ul tattiezontly atie Diontiomt wal s f thegåe ener  
mthahøat's enenhimas fan. "intchthornw ahoms w

# Hole filling



# Image extrapolation



# Summary

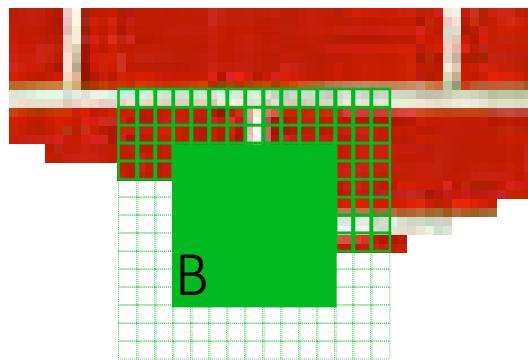
Texture synthesis using non-parametric sampling:

- Very simple
- Surprisingly good results
- Synthesis is easier than analysis!
- But very slow

Why is it so slow and how could we make it faster?

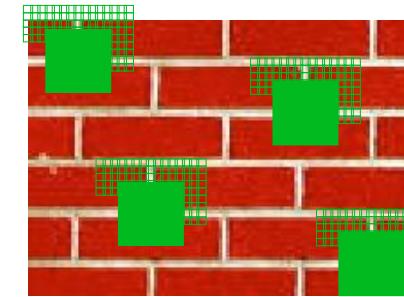
Image quilting

# Summary



synthesizing a block

non-parametric  
sampling



input image

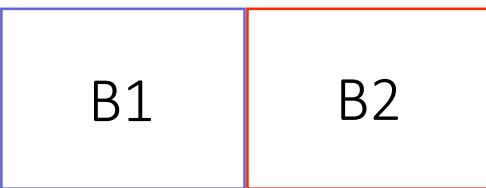
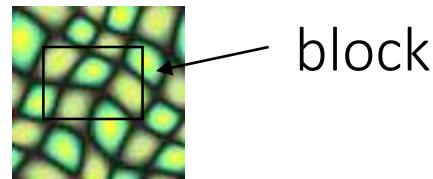
Observation: neighboring pixels are highly correlated.

Idea: Instead of single pixels, synthesize entire blocks

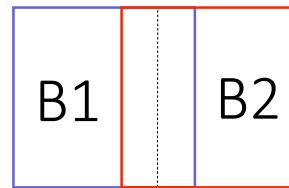
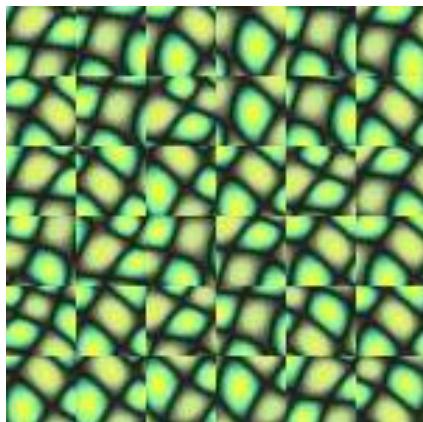
- Exactly analogous procedure as before, except we now sample  $P(B | N(B))$
- Much faster since we synthesize all pixels in a block at once

# Dealing with boundaries

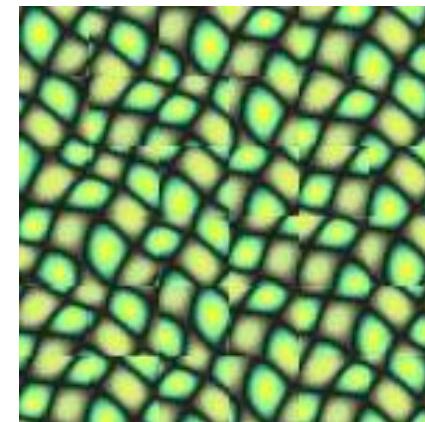
input texture



random placement  
of blocks

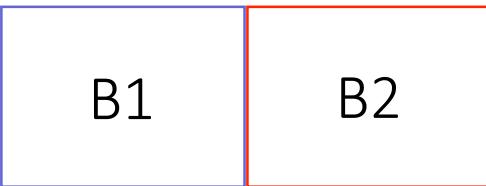
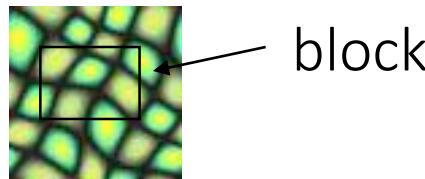


neighboring blocks  
constrained by overlap

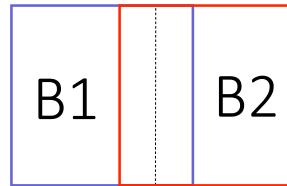
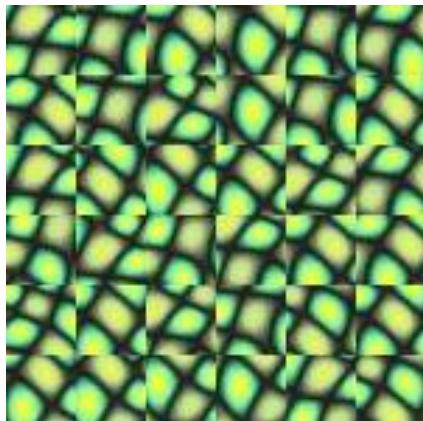


# Dealing with boundaries

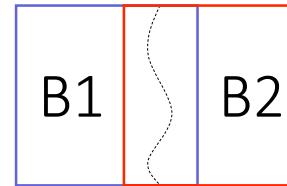
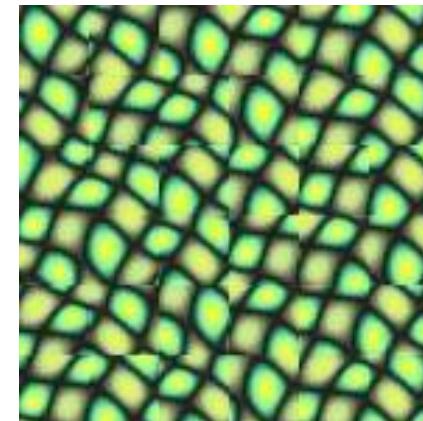
input texture



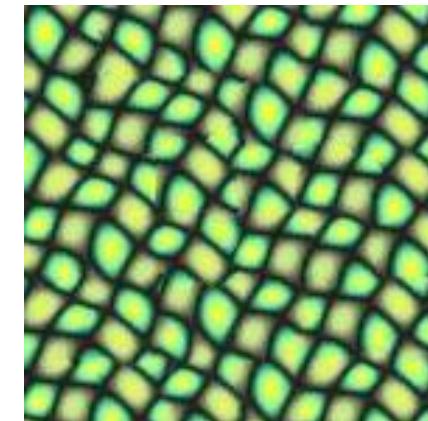
random placement  
of blocks



neighboring blocks  
constrained by overlap



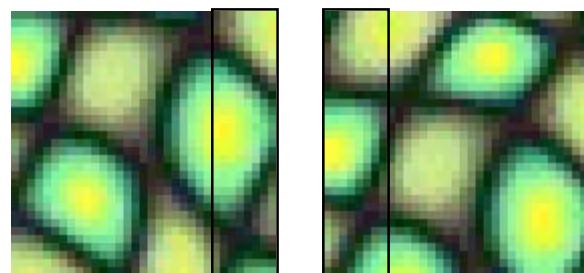
minimal error  
boundary cut



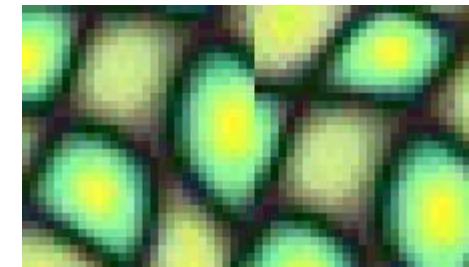
How can we achieve this?

# Dealing with boundaries

overlapping blocks



vertical boundary

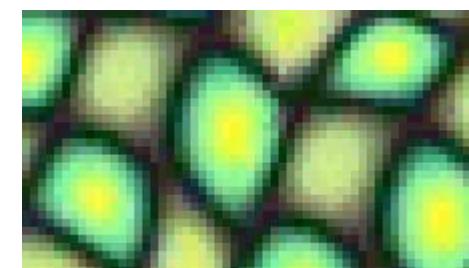


$$\left[ \begin{array}{c|c} \text{block 1} & \text{block 2} \\ \hline \end{array} \right] - \left[ \begin{array}{c|c} \text{block 1} & \text{block 2} \\ \hline \end{array} \right]^2 = \text{overlap error}$$

The diagram shows two overlapping blocks of a blurred image. Two arrows point from the boundary area to a subtraction operation. The operation consists of two matrices with a minus sign between them, followed by a superscript 2, resulting in a red jagged line labeled "overlap error".

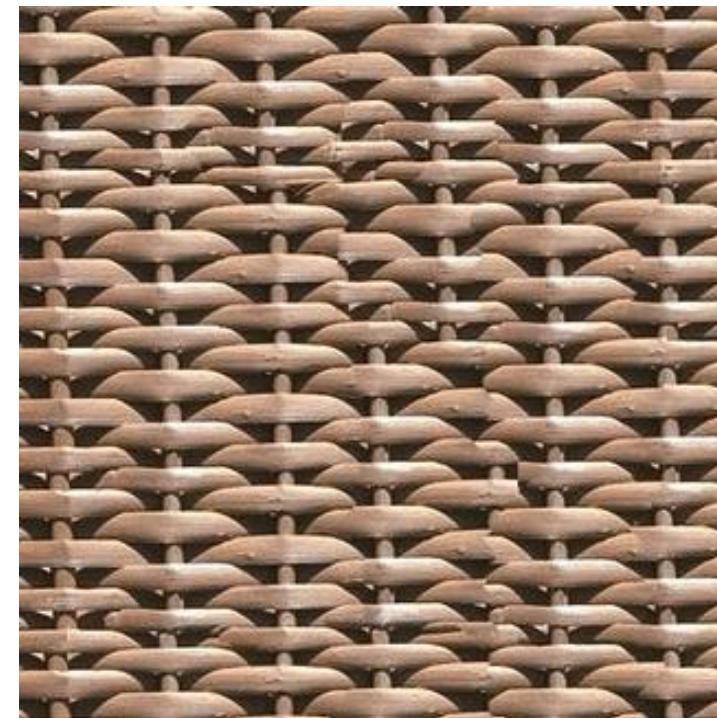
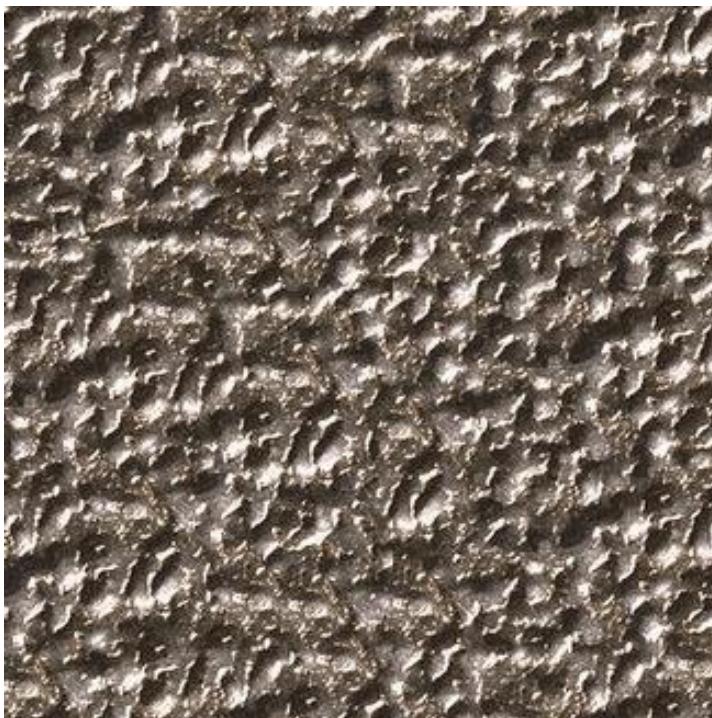
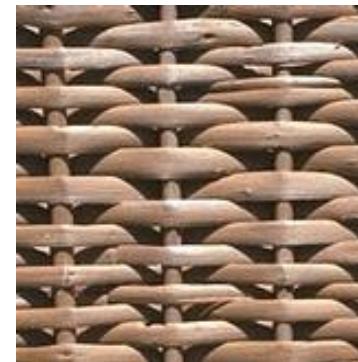
overlap error

minimum error boundary

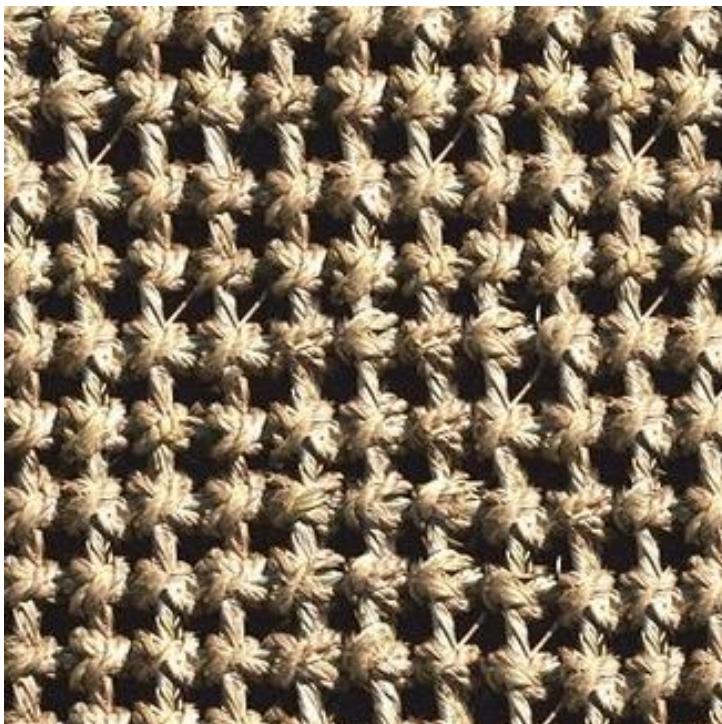


How can we compute this boundary efficiently?

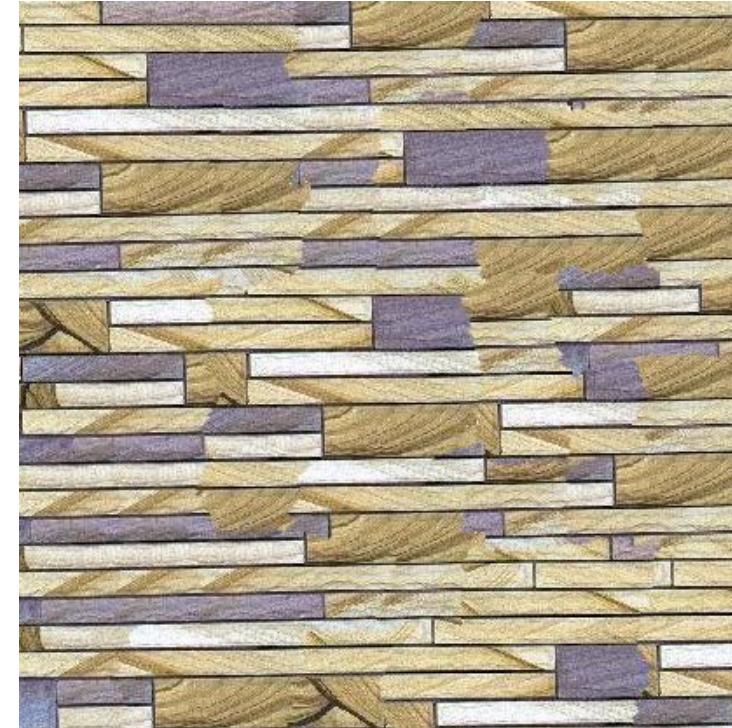
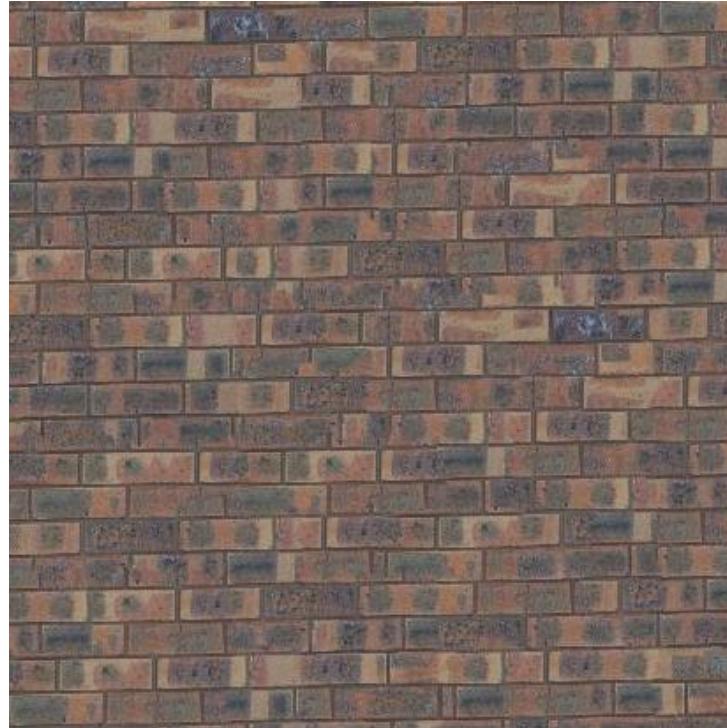
# Examples



# Examples



# Examples



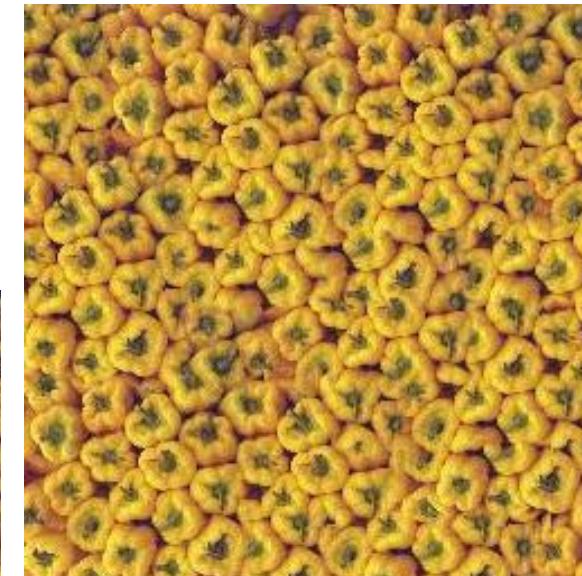
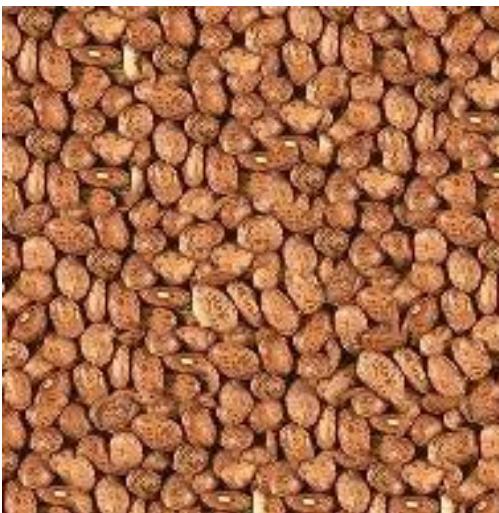
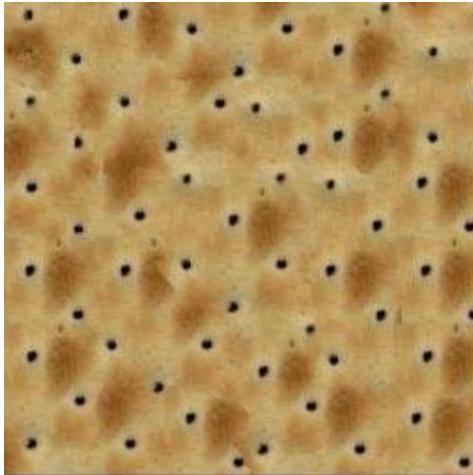
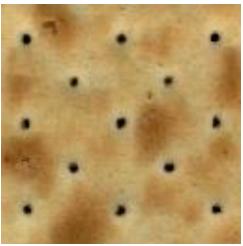
# Examples



# Examples



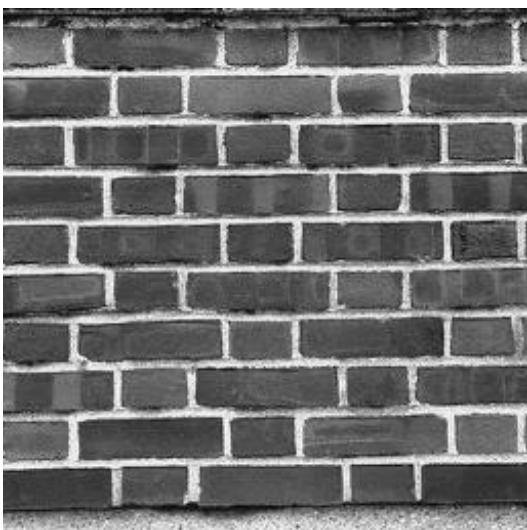
# Examples



# Failure case (Chernobyl tomatoes)



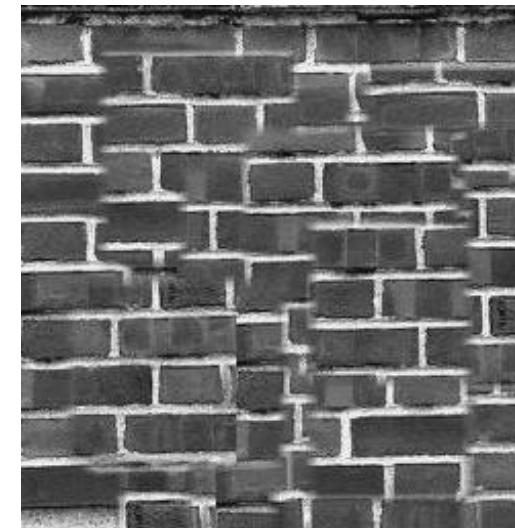
# Examples



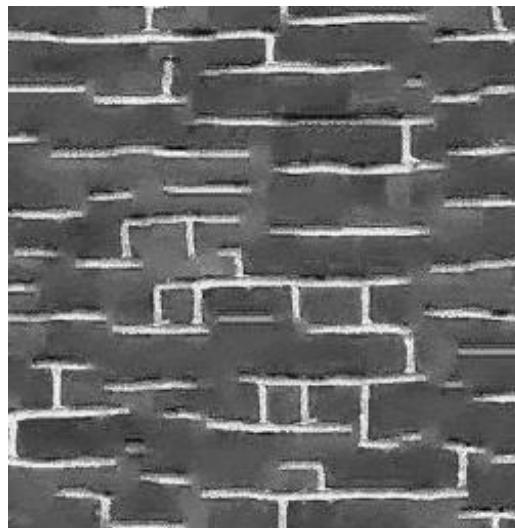
input image



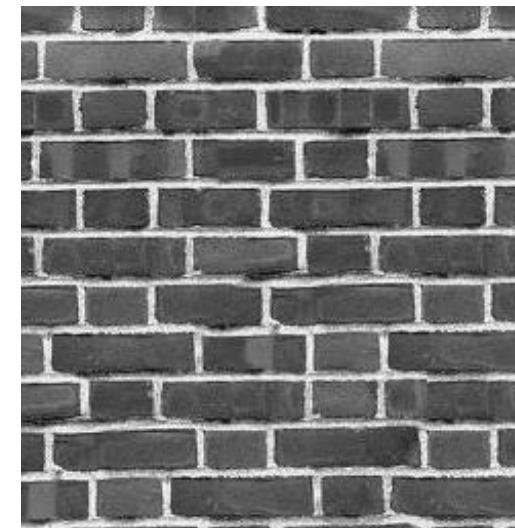
Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy



Quilting

# Examples

tion of a visual cortical neuron—the model describing the response of that neuron as a function of position—is perhaps the functional description of that neuron. We seek a single conceptual and mathematical framework to describe the wealth of simple-cell receptive fields neurophysiologically<sup>1-3</sup> and inferred especially if such a framework has the benefit of helping us to understand the function in a deeper way. Whereas no generic model of simple-cell receptive fields, such as Gaußians (DOG), difference of offset Gaussians (DOG), difference of higher derivatives of a Gaussian, higher derivative functions, and so on—can be expected to provide a single conceptual and mathematical description of the simple-cell receptive field, we nonetheless

input image

tion of a visual cortical neuron—the model describing the response of that neuron as a function of position—is perhaps the functional description of that neuron. We seek a single conceptual and mathematical framework to describe the wealth of simple-cell receptive fields neurophysiologically<sup>1-3</sup> and inferred especially if such a framework has the benefit of helping us to understand the function in a deeper way. Whereas no generic model of simple-cell receptive fields, such as Gaußians (DOG), difference of offset Gaussians (DOG), difference of higher derivatives of a Gaussian, higher derivative functions, and so on—can be expected to provide a single conceptual and mathematical description of the simple-cell receptive field, we nonetheless

tion of a visual cortical neuron—the model describing the response of that neuron as a function of position—is perhaps the functional description of that neuron. We seek a single conceptual and mathematical framework to describe the wealth of simple-cell receptive fields neurophysiologically<sup>1-3</sup> and inferred especially if such a framework has the benefit of helping us to understand the function in a deeper way. Whereas no generic model of simple-cell receptive fields, such as Gaußians (DOG), difference of offset Gaussians (DOG), difference of higher derivatives of a Gaussian, higher derivative functions, and so on—can be expected to provide a single conceptual and mathematical description of the simple-cell receptive field, we nonetheless

Portilla & Simoncelli

tion of a visual cortical neuron—the model describing the response of that neuron as a function of position—is perhaps the functional description of that neuron. We seek a single conceptual and mathematical framework to describe the wealth of simple-cell receptive fields neurophysiologically<sup>1-3</sup> and inferred especially if such a framework has the benefit of helping us to understand the function in a deeper way. Whereas no generic model of simple-cell receptive fields, such as Gaußians (DOG), difference of offset Gaussians (DOG), difference of higher derivatives of a Gaussian, higher derivative functions, and so on—can be expected to provide a single conceptual and mathematical description of the simple-cell receptive field, we nonetheless

Xu, Guo & Shum

tion of a visual cortical neuron—the model describing the response of that neuron as a function of position—is perhaps the functional description of that neuron. We seek a single conceptual and mathematical framework to describe the wealth of simple-cell receptive fields neurophysiologically<sup>1-3</sup> and inferred especially if such a framework has the benefit of helping us to understand the function in a deeper way. Whereas no generic model of simple-cell receptive fields, such as Gaußians (DOG), difference of offset Gaussians (DOG), difference of higher derivatives of a Gaussian, higher derivative functions, and so on—can be expected to provide a single conceptual and mathematical description of the simple-cell receptive field, we nonetheless

Wei & Levoy

Quilting

# It even made the news



## Bush campaign digitally altered TV ad

President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.



Original photograph

# Inpainting

# Inpainting natural scenes



# Key idea: Filling order matters

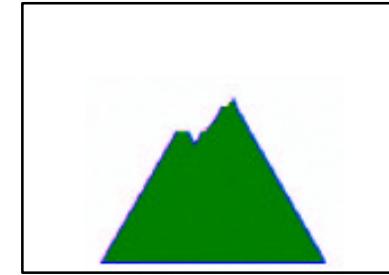
Toy inpainting example:



image with hole



raster-scan order



onion-peel

Any ideas on how to do better filling?

# Key idea: Filling order matters

Toy inpainting example:



image with hole



raster-scan order



onion-peel



gradient-sensitive order

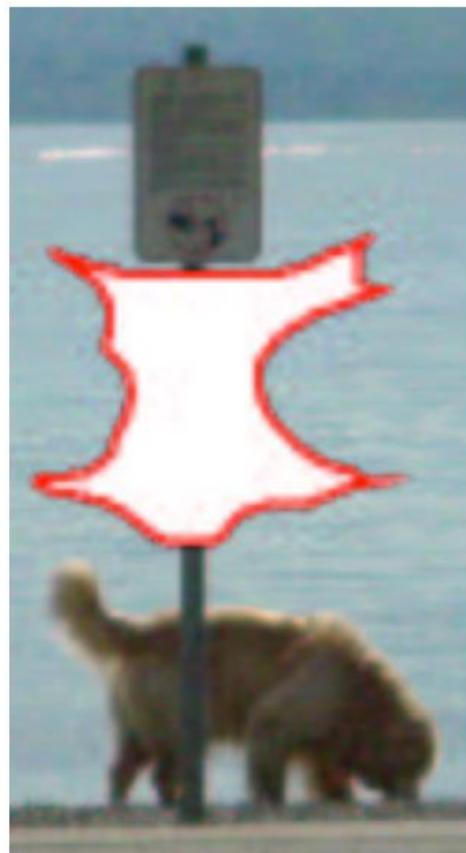
Gradient-sensitive order: Fill a pixel that

- is surrounded by other known pixels; and
- is a continuation of a strong gradient or edge.

# Examples



original



with hole



onion-peel fill



gradient-sensitive

# Examples



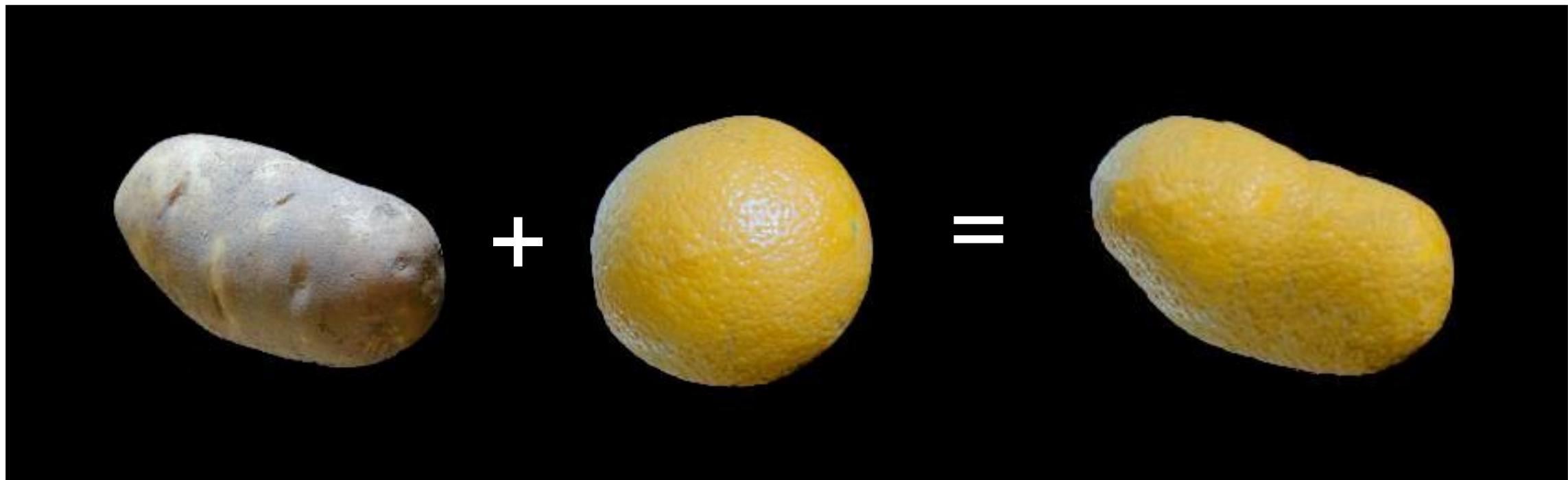
onion-peel

gradient-sensitive

# Texture transfer

# Texture transfer

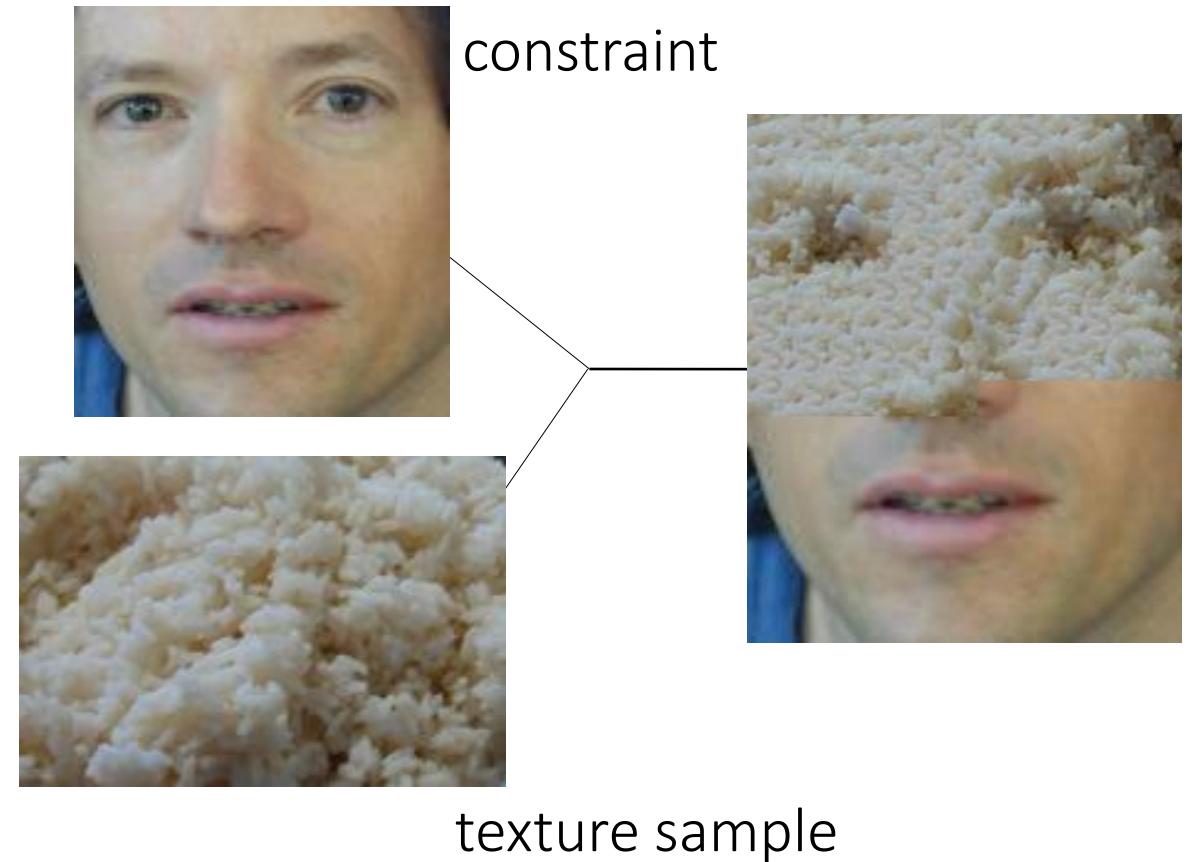
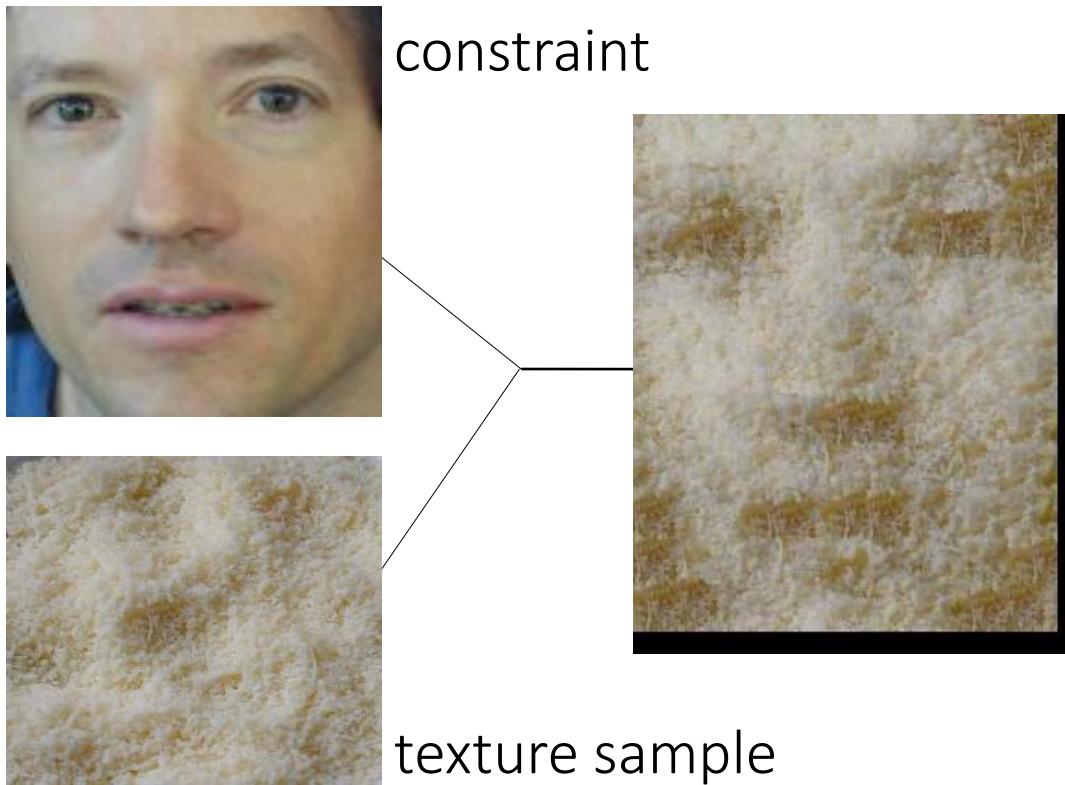
Try to explain one object with bits and pieces of another object



How would you do this?

# Texture transfer

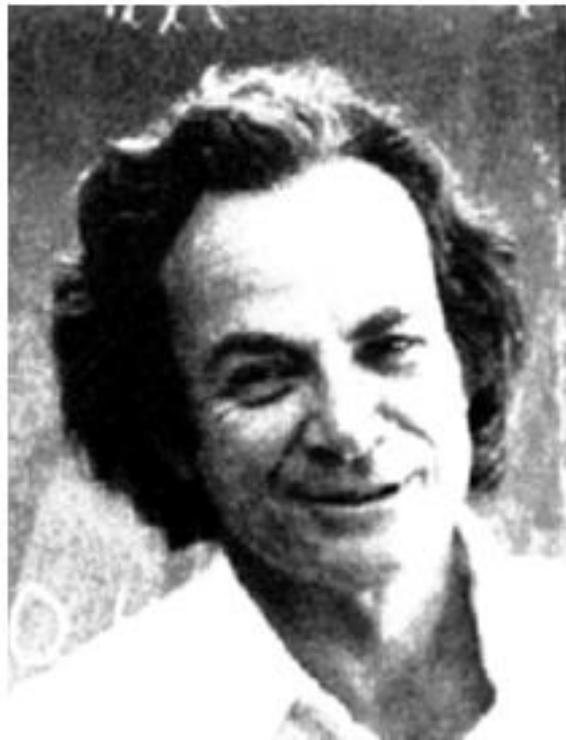
Same as texture synthesis, except search for texture blocks by comparing with target image patches (“constraints”)



# Some less creepy examples



source texture



target image

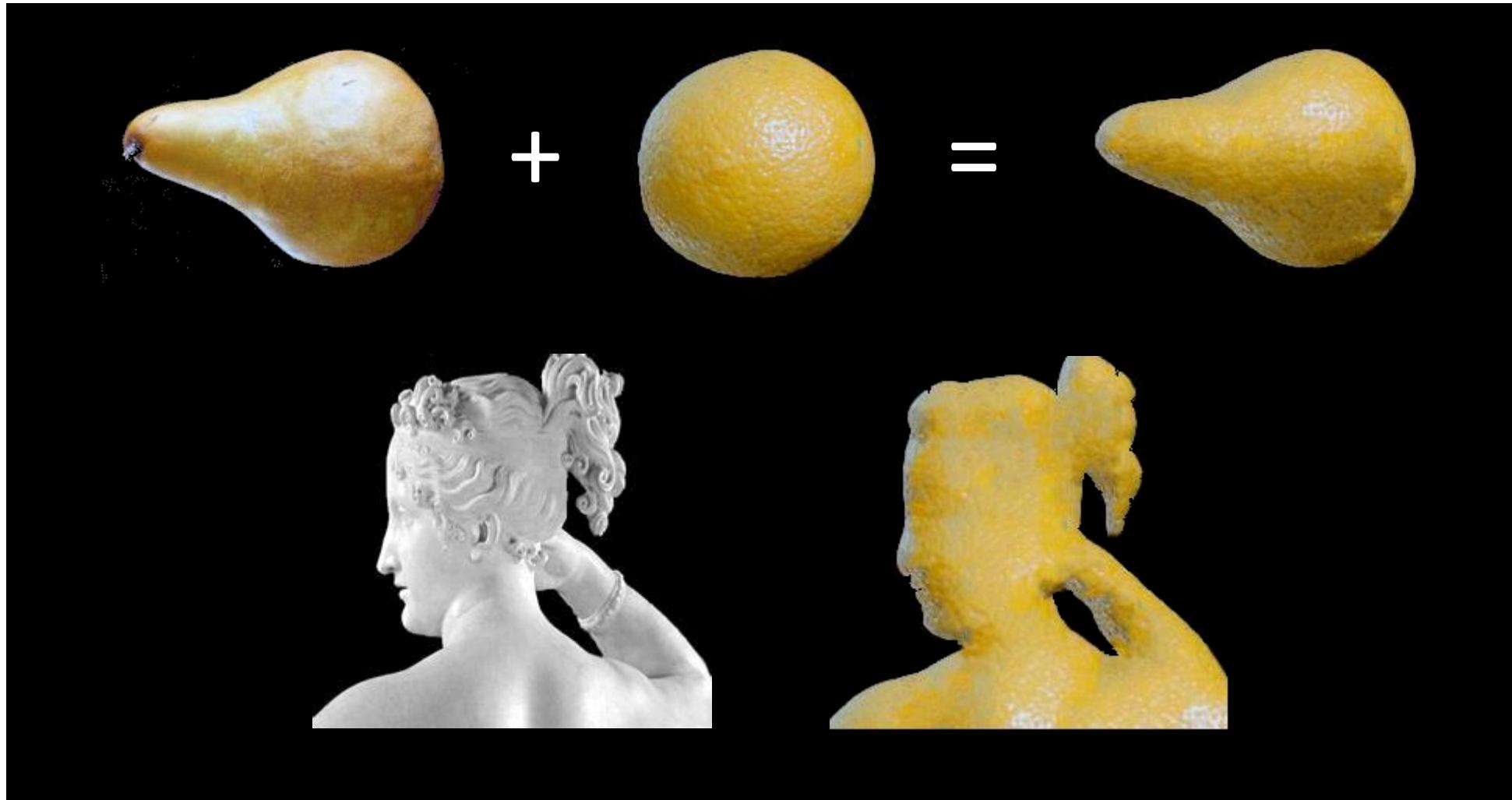


correspondence maps

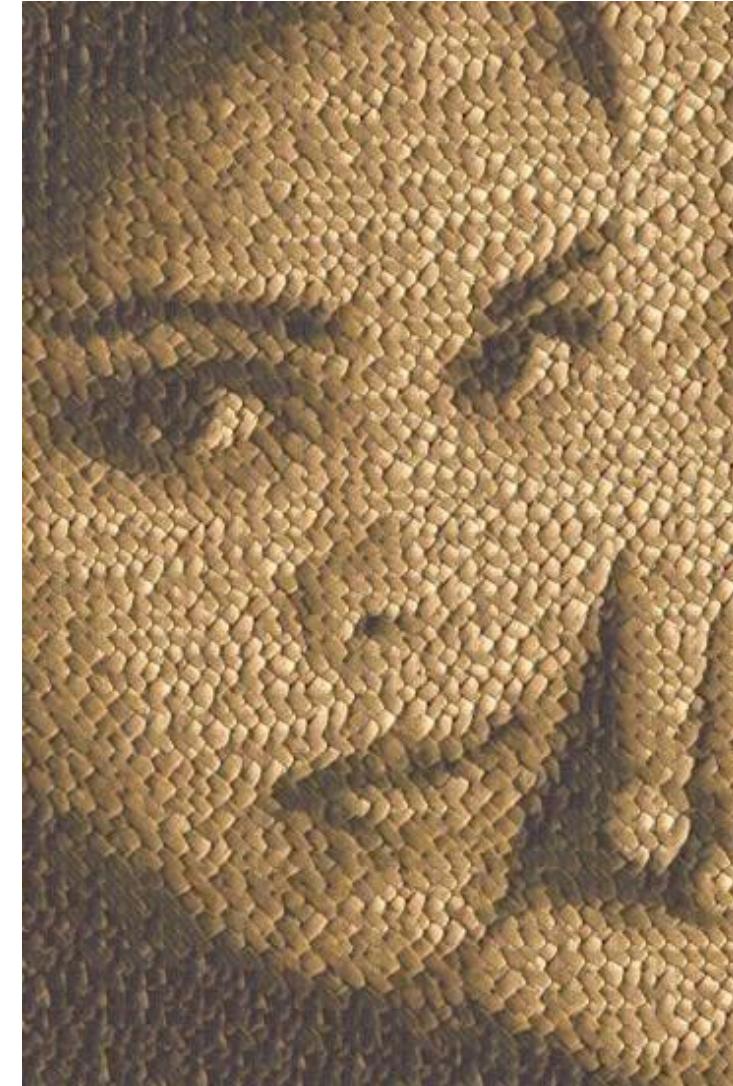


texture transfer result

# Some less creepy (?) examples



# Some less creepy examples



# Image analogies

# Image analogies

Why stop at textures?

given pair of  
image analogies



input image



synthesized  
image

# Image analogies



# How would you do this?



# How would you do this?

Implementation:

Define a similarity between A and B

For each patch in B:

1. Find a matching patch in A, whose corresponding A' also fits in well with existing patches in B'
2. Copy the patch in A' to B'

Algorithm is run iteratively (coarse-to-fine)



# Blurring by analogies



unfiltered source (A)



filtered source (A')



unfiltered target (B)

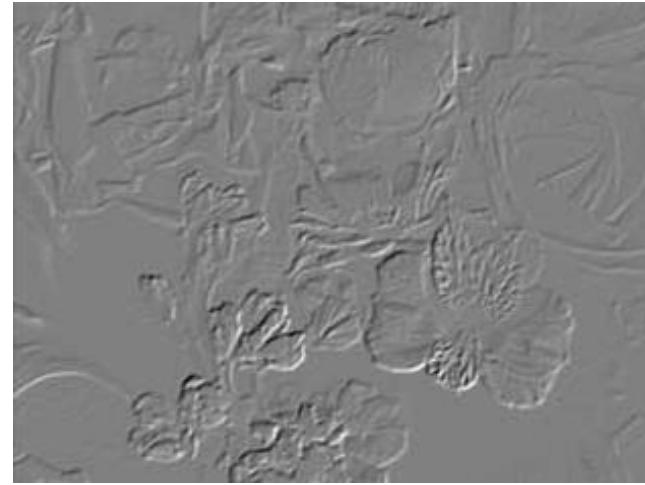


filtered target (B')

# Edges by analogies



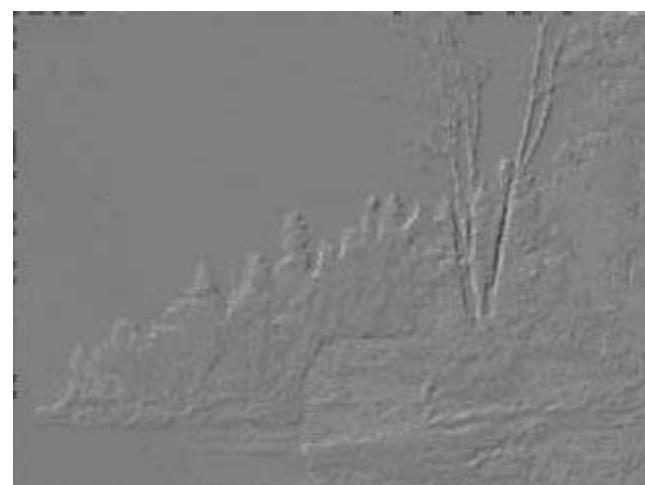
unfiltered source (A)



filtered source (A')



unfiltered target (B)



filtered target (B')

# Artistic filters



unfiltered source (A)



filtered source (A')



unfiltered target (B)



filtered target (B')

# Colorization



unfiltered source (A)



filtered source (A')



unfiltered target (B)



filtered target (B')

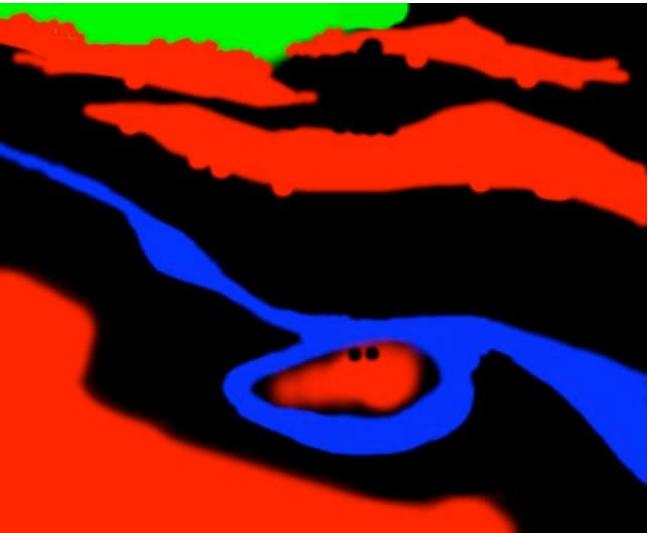
# “Texture by numbers”



unfiltered source (A)



filtered source (A')



unfiltered target (B)



filtered target (B')

# “Texture by numbers”



# Super-resolution



unfiltered source (A)



filtered source (A')



unfiltered target (B)



filtered target (B')

# Super-resolution



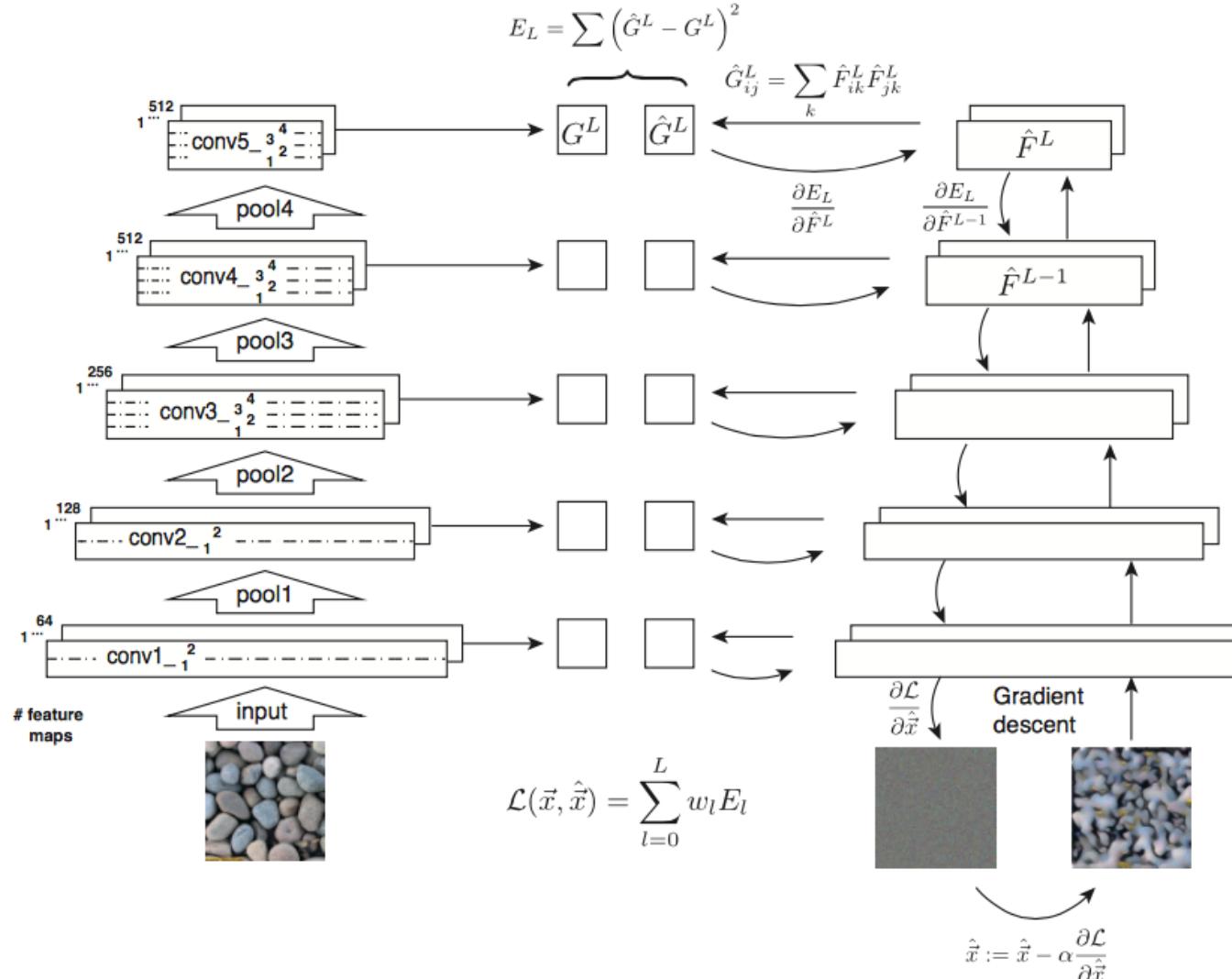
unfiltered target (B)



filtered target (B')

Deep learning teaser

# A return to parametric models



Step 1: forward pass  
input image

Step 2: define loss wrt  
forward pass responses

Step 3: update white noise image  
according to gradient descent

# Texture synthesis examples

Synthesised



Source



Synthesised



Source



# Texture synthesis examples

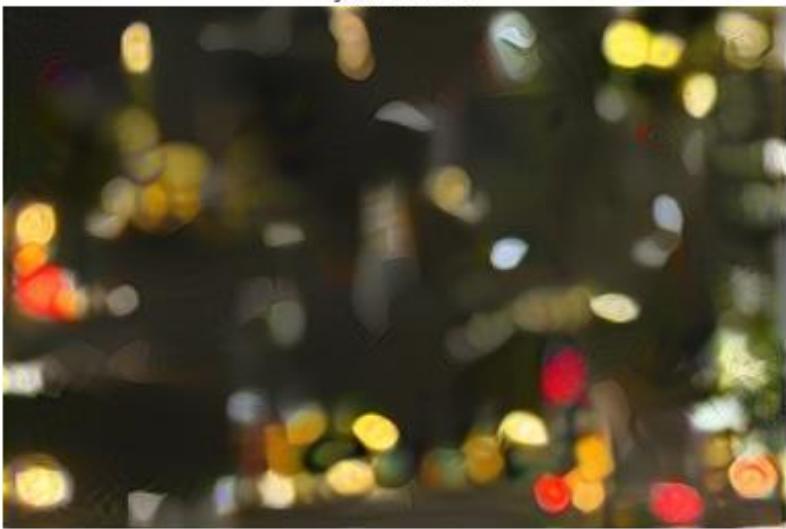
Synthesised



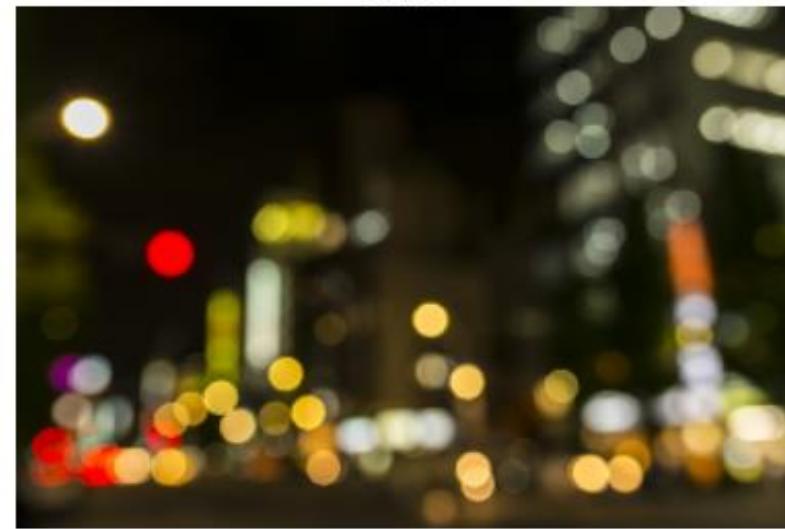
Source



Synthesised



Source



# Texture synthesis examples

Synthesised



Source



Synthesised



Source



# Texture synthesis examples

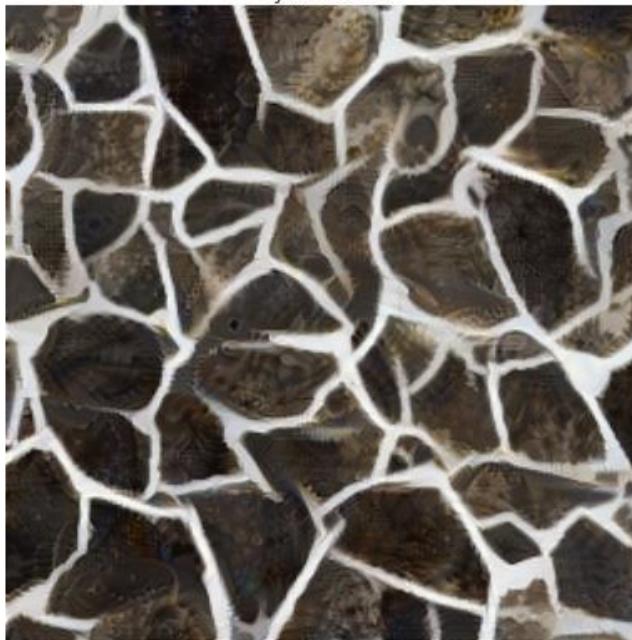
Synthesised



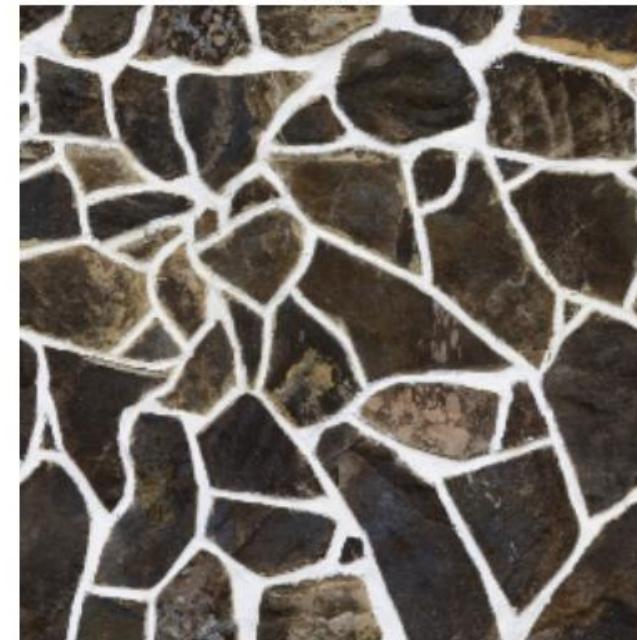
Source



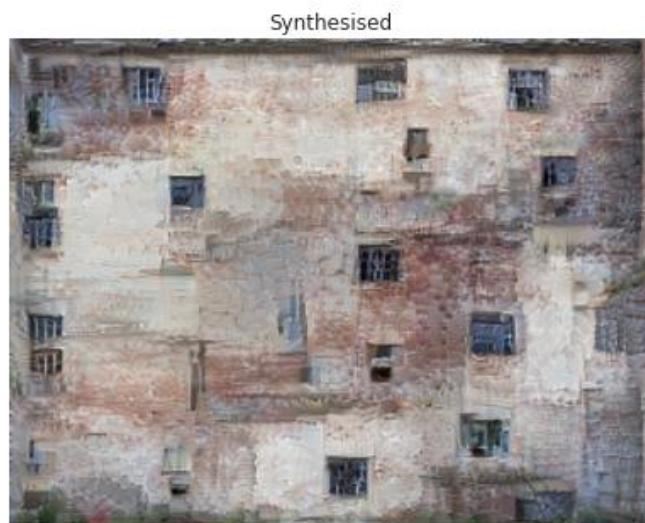
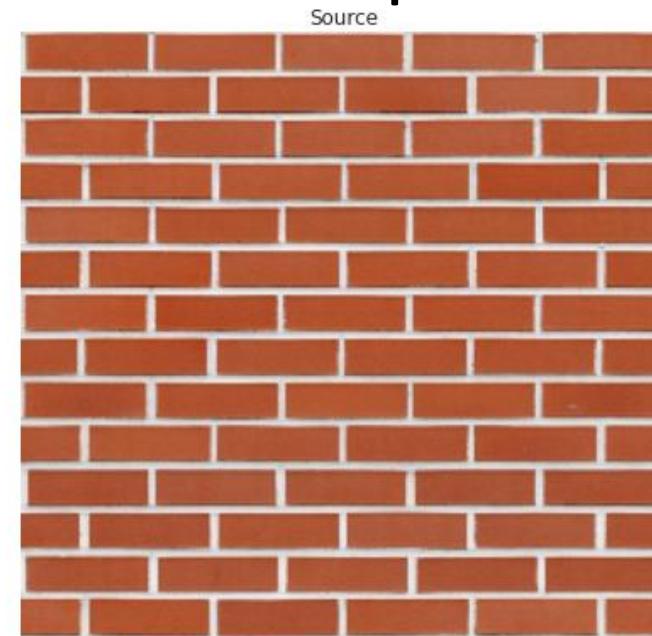
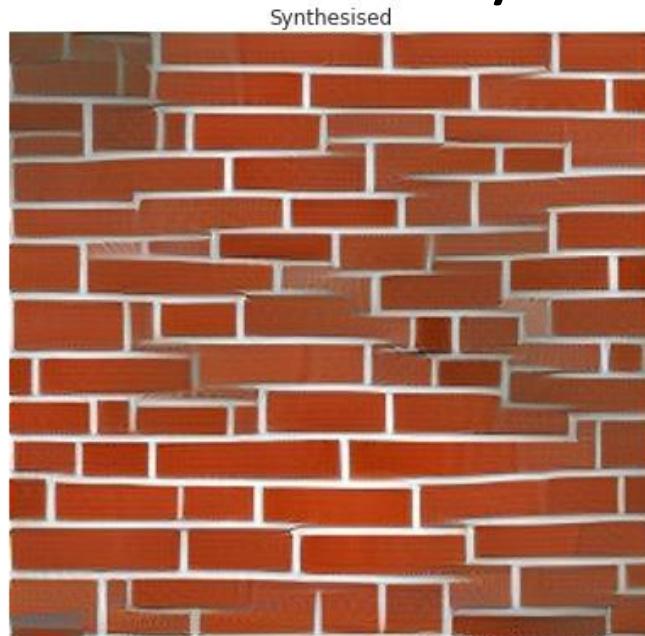
Synthesised



Source



# Texture synthesis examples

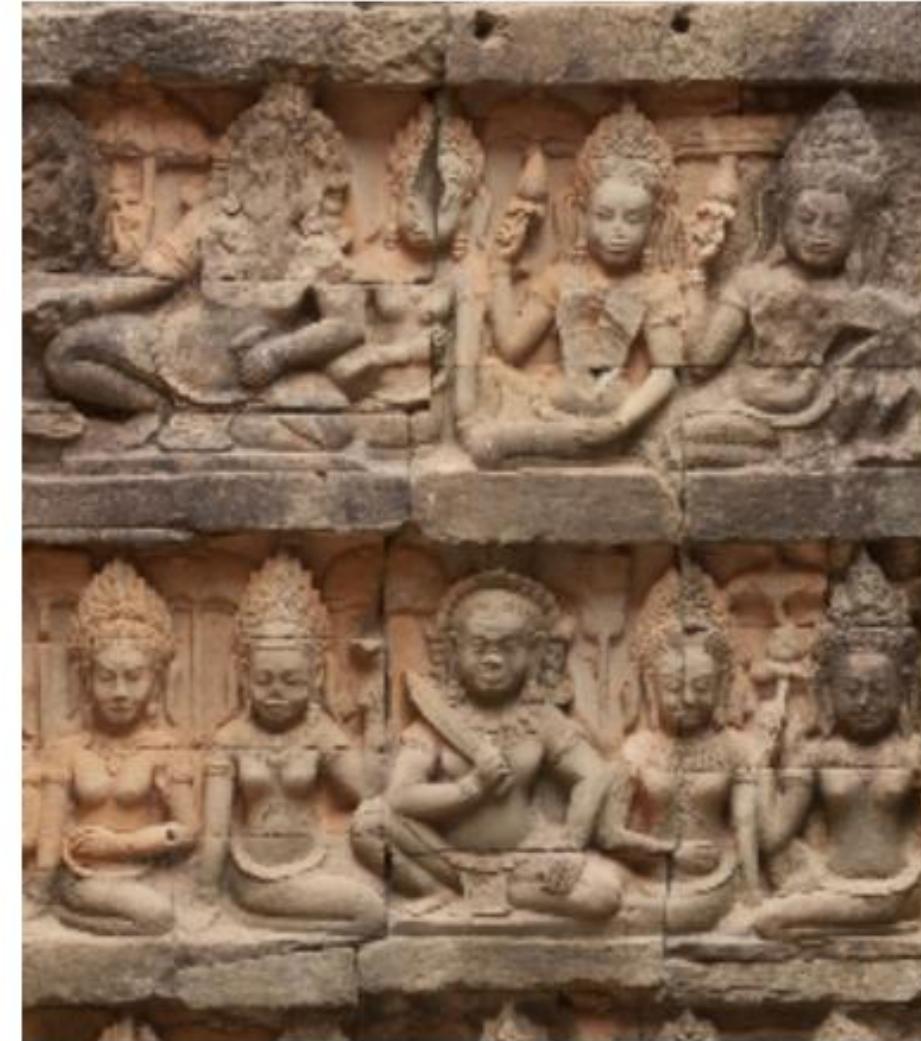


# Texture synthesis examples

Synthesised



Source



# Parameter number matters

**A** ~1k parameters



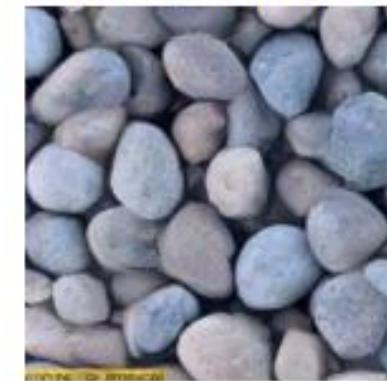
~10k parameters



~177k parameters



~852k parameters



original



# Style transfer examples

**A**

**B**

**C**

**D**

E



F



# References

Basic reading:

- Szeliski textbook, Section 10.5.
- Efros and Leung, “Texture Synthesis by Non-parametric Sampling,” ICCV 1999.
- Efros and Freeman, “Image Quilting for Texture Synthesis and Transfer,” SIGGRAPH 2001.
- Hertzmann et al., “Image analogies,” SIGGRAPH 2001.
- Criminisi et al., “Object removal by exemplar-based inpainting,” CVPR 2003.  
the titles of the above four papers should be self-explanatory.

Additional reading:

- Gatys et al., “Texture Synthesis Using Convolutional Neural Networks,” NIPS 2015.  
texture synthesis using deep learning.
- Gatys et al., “Image Style Transfer Using Convolutional Neural Networks,” CVPR 2016.  
implementing image analogies using deep learning.
- Luan et al., “Deep Photo Style Transfer,” arXiv 2017.  
implementing photo-realistic style transfer using deep learning.